Probabilistic Programming Semantics for Name Generation

MARCIN SABOK, McGill University, Canada
SAM STATON, University of Oxford, United Kingdom
DARIO STEIN, University of Oxford, United Kingdom
MICHAEL WOLMAN, McGill University, Canada

We make a formal analogy between random sampling and fresh name generation. We show that quasi-Borel spaces, a model for probabilistic programming, can soundly interpret the ν-calculus, a calculus for name generation. Moreover, we prove that this semantics is fully abstract up to first-order types. This is surprising for an 'off-the-shelf' model, and requires a novel analysis of probability distributions on function spaces. Our tools are diverse and include descriptive set theory and normal forms for the ν-calculus.

CCS Concepts: • **Theory of computation → Denotational semantics**; **Categorical semantics**; • **Mathematics of computing → Probability and statistics**.

Additional Key Words and Phrases: probabilistic programming, name generation, nu-calculus, quasi-Borel spaces, standard Borel spaces, descriptive set theory, Borel on Borel, denotational semantics, synthetic probability theory

ACM Reference Format:
Marcin Sabok, Sam Staton, Dario Stein, and Michael Wolman. 2021. Probabilistic Programming Semantics for Name Generation. *Proc. ACM Program. Lang.* 5, POPL, Article 11 (January 2021), 29 pages. https://doi.org/10.1145/3434292

1 INTRODUCTION

This paper is a foundational study of two styles of programming and their relationship:

(1) fresh name generation (gensym) via random draws;
(2) statistical probabilistic programming with higher-order functions.

We use a recent model of probabilistic programming, quasi-Borel spaces (QBSs, [Heunen et al. 2017]), to give a first random model of the ν-calculus [Pitts and Stark 1993], which is a λ-calculus with fresh name generation. By further developing the theory of QBSs, we are able to arrive at a new theorem for name generation:

Theorem (4.30)**.** *The random model of the ν-calculus is fully abstract at first order. That is, two first order programs are observationally equivalent if and only if their interpretation in QBSs is the same.*

This is surprising because the simple *non*-random models of the ν-calculus, based on nominal sets [Pitts 2013, Ch. 9.6] or functor categories [Stark 1996, §5], are *not* fully abstract at first order [Stark 1996, §5].

Authors' addresses: Marcin Sabok, Department of Mathematics and Statistics, McGill University, Montreal, Canada, marcin.sabok@mcgill.ca; Sam Staton, Department of Computer Science, University of Oxford, Oxford, United Kingdom, sam.staton@cs.ox.ac.uk; Dario Stein, Department of Computer Science, University of Oxford, Oxford, United Kingdom, dario.stein@cs.ox.ac.uk; Michael Wolman, Department of Mathematics and Statistics, McGill University, Montreal, Canada, michael.wolman@mail.mcgill.ca.

2475-1421/2021/1-ART11
https://doi.org/10.1145/3434292

1.1 The ν-Calculus and its Observational Equivalence

The ν-calculus (§2 and [Pitts and Stark 1993]) is a simply-typed λ-calculus with fresh name abstraction $\nu n.M$ in addition to λ-abstraction $\lambda x.M$. The idea is that $\nu n.M$ means "generate a fresh name n and continue as M". The ν-calculus thus models name generation as used in various domains across computer science, including cryptography, distributed systems, and statistical modelling (see §6 for more background on name generation). Concretely, the ν-calculus can also be viewed as a fragment of OCaml, where $\nu n.M$ abbreviates **let** n = ref () **in** M, since a content-less reference is a pure name when there is no pointer arithmetic or comparison allowed.

The purpose of this paper is to give an interpretation of name generation in terms of randomness. The ν-calculus already has a standard non-random operational semantics [Pitts and Stark 1993, §2], which induces a notion of observational equivalence $\approx$. For closed programs of ground type (name, bool), this is straightforward. For example, it includes the β/η laws of the call-by-value λ-calculus, and also equations such as

$$\nu m.\nu n.(m = n) \approx \texttt{false} \tag{1}$$

since any two separately generated names m, n should be different. Observational equivalence at first-order type (name $\to$ bool, bool $\to$ bool $\to$ name, etc.), on the other hand, is non-trivial in the ν-calculus, because ν's and λ's do not commute. For instance,

$$\nu n.\lambda x.n \not\approx \lambda x.\nu n.n. \tag{2}$$

So even at first order we can have complex nestings of ν's and λ's. In this paper we argue that a centerpiece of the first-order equational theory of the ν-calculus is the following 'privacy' equation [Pitts and Stark 1993, Ex. 4(2)]:

$$\nu n.\, \lambda x.(x = n) \quad \approx \quad \lambda x.\, \texttt{false} \quad : \mathsf{name} \to \mathsf{bool}. \tag{3}$$

On the left hand side, we generate a fresh name n, and then return a function that takes an argument x, and tests whether $x = n$. In this example, n is chosen to be different from any name that the caller of the function knows, and the name is never revealed to the caller, and so, intuitively, it can never return `true`. This is an example of an equation that is not validated by the standard nominal sets model, but it is validated by our QBS random model.

This aspect of name revelation is subtle, for instance, the program

$$\nu m.\, \nu n.\, \lambda x.\, \texttt{if}\ (x = m)\ \texttt{then}\ n\ \texttt{else}\ m \tag{4}$$

can reveal both m and n, but it needs to be called twice to do this. The random semantics takes care of this, as we explain.

1.2 Probabilistic Programming and Name Generation as Randomness

The idea of probabilistic programming (e.g. [van de Meent et al. 2018]) is to define complex probability distributions by writing programs. This is typically done by adding a sample command to a λ-calculus, to allow primitive random draws. In the statistical setting, it is common to include continuous distributions over the real numbers, such as the normal distribution (Fig. 1). For instance, the program

$$\textbf{let}\ \mathsf{x} = \mathsf{sample}(\mathsf{Normal}(0,1))\ \textbf{in}\ \textbf{let}\ \mathsf{y} = \mathsf{sample}(\mathsf{Normal}(0,1))\ \textbf{in}\ \mathsf{x}{+}\mathsf{y} \tag{5}$$

is overall equivalent to sampling from a normal (Gaussian) distribution with mean 0 and variance 2. The informal idea of this paper is to interpret $\nu n.M$ of the ν-calculus as a probabilistic program:

$$\text{“}\nu n.\, M \quad = \quad \textbf{let}\ \mathsf{n} = \mathsf{sample}(\mathsf{Normal}(0,1))\ \textbf{in}\ \mathsf{M}\text{”}$$

so that freshly generated names are randomly sampled. A first observation is that any two draws from a normal distribution will almost surely be different, and so this interpretation validates (1).

A probabilistic program involving sampling should be understood in terms of the histogram of results we see when we run the program a large number of times. To put it another way, the program (5) is a Monte Carlo description of the integral $\iint k(x+y)\,\mathrm{d}y\,\mathrm{d}x$ where $\int$ denotes Lebesgue integration with respect to the normal probability measure and k is some continuation function. In this way, we may say, informally for now, that the random implementation of ν-abstraction is also Lebesgue integration:

$$\text{“}\nu n.\, M \;=\; \int M\,\mathrm{d}n\text{”}$$

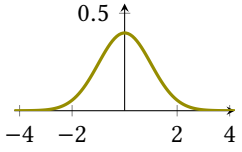

Fig. 1. Density of the normal distribution Normal(0,1).

As we will make precise in Sections 1.3 and 3.3, the measure-theoretic understanding of probability leads to full abstraction at first order. For a first glimpse, notice that in the ν-calculus there is no definable function

$$\exists : (\mathsf{name} \to \mathsf{bool}) \to \mathsf{bool} \tag{6}$$

such that $\exists(f)$ returns `true` if f would ever return `true`, as such a function would easily distinguish the programs in the privacy equation (3). This function $\exists$ *can* be defined in the nominal sets model (e.g. [Pitts 2013, §2.5], [Staton 2010, eq. 2]), but is *in*consistent with a measure-theoretic interpretation, as we now explain. From this function $\exists$ we could easily define an expression

$$f : \mathsf{name} \to \mathsf{name} \to \mathsf{bool} \;\vdash\; \lambda x.\, \exists(\lambda y.\, f\, x\, y) \;\; : \mathsf{name} \to \mathsf{bool}$$

which converts a subset of $(\mathsf{name} \times \mathsf{name})$ to its existential projection as a subset of (name). In the setting of probability theory, we need to know that all definable expressions are measurable, so that integration can be used. If we understand (name) as the real numbers, and measurable subsets are Borel sets, as usual, then the projection of a Borel set is not necessarily Borel [Kechris 1987, 14.2], and so the $\exists$ function (6) cannot be in the model. So our probabilistic interpretation of the ν-calculus gives a new intuition for these privacy and definability issues.

1.3 Quasi-Borel Spaces, Full Abstraction and Descriptive Set Theory

A formalism that includes both measure theory and typed λ-calculus is quasi-Borel spaces (QBSs, §3.2 and [Heunen et al. 2017]). A QBS is a set X together with a set of functions $M_X \subseteq [\mathbb{R} \to X]$ satisfying some conditions. The idea is to fix $\mathbb{R}$ as a source of randomness, and then M_X describes the admissible random elements in X. For example, for the QBS of booleans, we take $M_{\mathsf{bool}} \subseteq [\mathbb{R} \to 2]$ to comprise the characteristic functions of Borel sets of $\mathbb{R}$, and for the QBS function space $[\mathsf{real} \to \mathsf{bool}]$ we take $M_{\mathsf{real}\to\mathsf{bool}} \subseteq [\mathbb{R} \to (\mathbb{R} \to 2)]$ to comprise the characteristic functions of Borel subsets of $\mathbb{R}^2$. In this way, we can interpret any ν-calculus type as a QBS (§3.3). Following the above discussion, we see that $\exists$ (6) cannot be interpreted in QBSs.

We show our full abstraction theorem in this setting: two ν-calculus programs of first-order type are observationally equivalent if and only if their interpretations in QBSs are equal (Thm. 4.30). Our proof proceeds in three steps.

(1) We show that the privacy equation (3) holds in QBS (§4). We have already mentioned that the $\exists$ function (6) cannot be defined in QBSs. The next step is to fully characterize the QBS space corresponding to $((\mathsf{name} \to \mathsf{bool}) \to \mathsf{bool})$. This turns out to correspond directly with the concept of ‘Borel-on-Borel’ in descriptive set theory [Kechris 1987, §18.B], and we use a pair of Borel inseparable sets to generalize the non-definability of $\exists$ and prove the privacy equation (Thm. 4.1).

$$\sigma, \tau ::= \mathsf{B} \mid \mathsf{N} \mid \sigma \to \tau \qquad (\mathsf{B} \text{ and } \mathsf{N} \text{ abbreviate (bool) and (name) from §1 respectively.})$$
$$M, N ::= x \mid \mathtt{true} \mid \mathtt{false} \mid M = M \mid MM \mid \lambda x.M \mid \nu n.M \mid \mathtt{if}\ M\ \mathtt{then}\ M\ \mathtt{else}\ M$$

$$\frac{}{\Gamma \vdash x : \tau}\ ((x:\tau) \in \Gamma) \qquad \frac{}{\Gamma \vdash b : \mathsf{B}}\ (b = \mathtt{true}, \mathtt{false})$$

$$\frac{\Gamma \vdash M : \mathsf{B} \quad \Gamma \vdash N_1 : \tau \quad \Gamma \vdash N_2 : \tau}{\Gamma \vdash \mathtt{if}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2 : \tau} \qquad \frac{\Gamma \vdash M : \mathsf{N} \quad \Gamma \vdash N : \mathsf{N}}{\Gamma \vdash (M = N) : \mathsf{B}}$$

$$\frac{\Gamma, x : \mathsf{N} \vdash M : \tau}{\Gamma \vdash \nu x.M : \tau} \qquad \frac{\Gamma, x : \sigma \vdash M : \tau}{\Gamma \vdash \lambda x.M : \sigma \to \tau} \qquad \frac{\Gamma \vdash M : \sigma \to \tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash M\,N : \tau}$$

Fig. 2. Grammar and typing rules for the ν-calculus [Pitts and Stark 1993, Table 1].

(2) On the syntactic side, we give a normalization algorithm for observational equivalence at first order (§4.2, Thm. 4.25). Our algorithm, which appears to be novel, refines a logical relations argument by Pitts and Stark [Pitts and Stark 1993], by identifying and eliminating all private names. This is non-trivial as, for instance, (4) is already in normal form, but the similar program

$$\nu m.\, \nu n.\, \lambda x.\, \mathtt{if}\ (x = m)\ \mathtt{then}\ m\ \mathtt{else}\ n \qquad \text{normalizes to} \qquad \nu n.\, \lambda x.\, n.$$

Our construction simplifies the analysis of observational equivalence at first order (Thm. 4.25). This also provides a general strategy for proving full abstraction (Thm. 4.26).

(3) Returning to the semantic side, we show that the normalization steps are validated in the QBS model (§4.3). The key idea here is that atomless measures such as the normal and uniform distributions are invariant under certain translations. We use this translation invariance to reduce our problem to the privacy equation (3), and use this to prove full abstraction at first order (Thm. 4.30). Our use of an invariant action on the space of names is similar to but distinct from nominal techniques [Pitts 2013, §1.9]; our action is internal to the model, and does not feature in its construction.

In addition to proving full abstraction of the QBS semantics of the ν-calculus at first order, we provide the first detailed investigation of the higher-typed function spaces in Borel-based probability theory (§4.1, §5). The application of higher-order probabilistic methods is increasingly widespread in programming research (§6.3 and [Ehrhard et al. 2018; Lew et al. 2019; Sato et al. 2019; Ścibior et al. 2017; Vandenbroucke and Schrijvers 2020]). We show that our programming-based development can alternatively be viewed in terms of recent categorical formulations of probability theory (§5). From this perspective, Bayesian inference (conditioning) is subtle in the higher-typed situation (Prop. 5.2). Intuitively, arbitrary conditioning would mean that one could infer, from data as a function (name $\to$ bool), a posterior distribution on the names that the function privately uses, in violation of the privacy equation (3).

In summary, through our full abstraction result (Thm. 4.30), we formalize the relationship between random sampling and fresh name generation, giving new perspectives on higher-order probability.

2 PRELIMINARIES ON NAME GENERATION AND THE ν-CALCULUS

In this section we recall the ν-calculus [Pitts and Stark 1993; Stark 1994], which is a simple λ-calculus for name generation. We recall the syntax, the observational equivalence (§2.1) and the denotational semantics (§2.2). Further discussion about name generation is in Section 6.

The types, syntax and typing judgements of the ν-calculus are recalled in Fig. 2 [Pitts and Stark 1993]. The typing judgements are of the form $\Gamma \vdash M : \tau$, where Γ is a set of typed variables.

$$\frac{-}{s \vdash V \Downarrow_\tau ()V} \qquad \frac{s \vdash M \Downarrow_{\mathsf{N}} (s_1)m \quad s \vdash N \Downarrow_{\mathsf{N}} (s_2)n}{s \vdash (M = N) \Downarrow_{\mathsf{B}} (s_1 \uplus s_2)\mathtt{false}}\ m \neq n \qquad \frac{s \vdash M \Downarrow_{\mathsf{N}} (s_1)m \quad s \vdash N \Downarrow_{\mathsf{N}} (s_2)m}{s \vdash (M = N) \Downarrow_{\mathsf{B}} (s_1 \uplus s_2)\mathtt{true}}$$

$$\frac{s \vdash M \Downarrow_{\mathsf{B}} (s_1)V \quad (s \uplus s_1) \vdash N_V \Downarrow_\tau (s_2)V'}{s \vdash \mathtt{if}\ M\ \mathtt{then}\ N_{\mathtt{true}}\ \mathtt{else}\ N_{\mathtt{false}} \Downarrow_\tau (s_1 \uplus s_2)V'} \qquad \frac{s \uplus \{n\} \vdash M \Downarrow_\tau (s')V}{s \vdash \nu n.\, M \Downarrow_\tau (\{n\} \uplus s')V}\ n \notin s$$

$$\frac{s \vdash M \Downarrow_{\sigma\to\tau} (s_1)\lambda x.M' \quad (s \uplus s_1) \vdash N \Downarrow_\sigma (s_2)V \quad (s \uplus s_1 \uplus s_2) \vdash M'[V/x] \Downarrow_\tau (s_3)V'}{s \vdash M\,N \Downarrow_\tau (s_1 \uplus s_2 \uplus s_3)V'}$$

Fig. 3. Evaluation relation for the ν-calculus [Pitts and Stark 1993, Table 2].

The types B, N are called *ground types*. Among higher types, we will pay special attention to *first-order* function types, which are non-nested function types of the form $\tau_1 \to \cdots \to \tau_n$ with each τ_i a ground type. As the ν-calculus is call-by-value, first-order function types cannot be simplified by uncurrying and already contain considerable complexity. We elaborate this at the end of (§2.2).

2.1 Operational Semantics and Observational Equivalence

The evaluation relation of the ν-calculus is defined for terms with free variables of type N, and no other free variables. In this operational semantics, these variables are understood to be names that are generated in the course of running a program, and so they are assumed to be distinct, and we tend to use m or n for them. If $s = \{n_1, \ldots, n_k\}$ is a set of names and τ is a type, we define a set

$$\mathrm{Exp}_\tau(s) \stackrel{\text{def}}{=} \left\{ M \mid n_1 \colon \mathsf{N}, \ldots n_k \colon \mathsf{N} \vdash M : \tau \right\}$$

of expressions of type τ only involving the names s, and we define the set $\mathrm{Val}_\tau(s) \subseteq \mathrm{Exp}_\tau(s)$ of values: $\mathrm{Val}_\tau(s) = \{V \in \mathrm{Exp}_\tau(s) \mid V = \lambda x.M,\ V = \mathtt{true},\ V = \mathtt{false},\ V = n\}$.

If s, t are sets of names, we write $s \uplus t$ to denote the *disjoint union* of these names, which we can always form by renaming free names if necessary.

The big-step evaluation relation $s \vdash M \Downarrow_\tau (s')V$ is given in Figure 3, where $M \in \mathrm{Exp}_\tau(s)$ and $V \in \mathrm{Val}_\tau(s \uplus s')$, meaning M evaluates to V generating fresh names s'. Evaluation is terminating and deterministic up to choice of free names. (We will not need to work directly with this evaluation relation very much in this paper, because we will build on existing methods for observational equivalence [Pitts and Stark 1993; Stark 1996], but we include it for completeness.)

Observational equivalence is defined in a standard way. A *boolean context* $C[\cdot]$ *for type* τ is an expression C where some subexpressions are replaced by a placeholder, such that if $M \in \mathrm{Exp}_\tau(s)$ then $C[M] \in \mathrm{Exp}_{\mathsf{B}}(s)$. Two terms $M_1, M_2 \in \mathrm{Exp}_\tau(s)$ are *observationally equivalent*, written $M_1 \approx_\tau M_2$, if for every boolean context $C[\cdot]$ we have $\exists s'(s \vdash C[M_1] \Downarrow_{\mathsf{B}} (s')\mathtt{true})$ if and only if $\exists s'(s \vdash C[M_2] \Downarrow_{\mathsf{B}} (s')\mathtt{true})$.

We have already given some examples of observational equivalences and inequivalences in Section 1.1. We illustrate the method a little more. To see that $\nu n.\lambda x.n \not\approx_{\mathsf{B}\to\mathsf{N}} \lambda x.\nu n.n$ (2), consider the context $C[-] = (\lambda f.(f\,\mathtt{true}) = (f\,\mathtt{true}))\,(-)$, which produces `true` for the first example and `false` for the right hand side. On the other hand, an observational equivalence such as $\nu n.\lambda x.(x = n) \approx_{\mathsf{N}\to\mathsf{B}} \lambda x.\mathtt{false}$ (3) is a statement that quantifies over all contexts, and so requires a more elaborate method such as logical relations [Pitts and Stark 1993, Example 5] or our random model (§4.1).

We remark that the call-by-value semantics of the ν-calculus form a central aspect of the intricacies of observational equivalence at first-order types. The $\lambda\nu$-calculus is a call-by-*name* variation of the ν-calculus [Odersky 1994; Pitts 2013, §9.4], and in that calculus, λ's and ν's do commute

[Odersky 1994, Fig. 2], and then we can easily derive

$$\nu n.\lambda x.(x = n) \approx_{\mathsf{N}\to\mathsf{B}} \lambda x.\nu n.(x = n) \approx_{\mathsf{N}\to\mathsf{B}} \lambda x.\mathtt{false}. \tag{7}$$

2.2 Categorical Semantics

The central definition of this paper is the random semantics of the ν-calculus in Section 3.3. Although this is a new semantics for the ν-calculus, it is an instance of the very general categorical framework for ν-calculus semantics given by Stark [Stark 1996]. The rough idea is that one can interpret the ν-calculus in any category with enough structure, by interpreting types as objects of the category and expressions as morphisms.

Metalanguages. This interpretation is clarified by using a metalanguage (aka internal language) to describe the morphisms of the category, and the way that they compose, instead of the traditional categorical composition notation (e.g. [Lambek and Scott 1988, §I.10]). The metalanguage of cartesian closed categories allows us to notate a morphism $A_1 \times \cdots \times A_n \to B$ as an expression $x_1 : A_1 \dots x_n : A_n \vdash e : B$, and to use λ-notation and pairing to manipulate the function spaces and products in the category. Where the category also has a coproduct $1 + 1$, we can write the injections as $\vdash \mathtt{true} : 1 + 1$ and $\vdash \mathtt{false} : 1 + 1$, and the universal property of coproducts can be expressed in terms of an `if / then / else` construction. The interpretation of the ν-calculus in a categorical model can be given by a translation from the ν-calculus to this metalanguage.

Commutative Affine Monads. A strong monad $(T, [-], (-)^*)$ on a cartesian closed category $\mathbb{C}$ comprises an assignment of an object $T(A)$ for every object A in $\mathbb{C}$, a family of 'return' morphisms $[-] : A \to T(A)$, and a family of 'bind' operations $(-)^* : T(B)^A \to T(B)^{T(A)}$, satisfying associativity and identity laws [Moggi 1991]. In terms of the metalanguage, for any morphisms described by expressions $\Gamma \vdash e : T(A)$ and $\Gamma, x : A \vdash e' : T(B)$, we have a morphism described by an expression $\Gamma \vdash \mathtt{let}\ x \leftarrow e\ \mathtt{in}\ e' : T(B)$ [Moggi 1991]. A strong monad is called *affine* and *commutative* if the following *discardability* (8) and *exchangeability* (9) equations in the metalanguage are valid:

$$\mathtt{let}\ x \leftarrow e\ \mathtt{in}\ e' \ = \ e' \qquad (x \text{ not free in } e') \tag{8}$$

$$\mathtt{let}\ x_1 \leftarrow e_1\ \mathtt{in}\ \mathtt{let}\ x_2 \leftarrow e_2\ \mathtt{in}\ e_3 \ = \ \mathtt{let}\ x_2 \leftarrow e_2\ \mathtt{in}\ \mathtt{let}\ x_1 \leftarrow e_1\ \mathtt{in}\ e_3 \qquad (x_1 \text{ not free in } e_2,\ x_2 \text{ not free in } e_1). \tag{9}$$

Informally, affine means that we can discard any unused expressions, and is equivalent to $T(1) \cong 1$. Commutativity means that we can exchange independent expressions (e.g. [Kammar and Plotkin 2012]).

Definition 2.1 ([Stark 1996, §4.1])**.** A *categorical model of the ν-calculus* comprises

(1) a cartesian closed category $\mathbb{C}$ with finite limits;
(2) a strong monad T on $\mathbb{C}$;
(3) a disjoint coproduct $B := 1 + 1$ of the terminal object with itself;
(4) a distinguished object of names N with a decidable equality test $(=) : N \times N \to B$; and
(5) a distinguished morphism $\mathsf{new} : 1 \to T(N)$.

We ask that this category satisfies the following additional axioms:

(1) the monad T is affine and commutative;
(2) the following equation holds in the metalanguage

$$m : N \vdash \mathtt{let}\ n \leftarrow \mathsf{new}\ \mathtt{in}\ [(n, m = n)] \ = \ \mathtt{let}\ n \leftarrow \mathsf{new}\ \mathtt{in}\ [(n, \mathtt{false})] : T(N \times B). \tag{FRESH}$$

The (FRESH) requirement allows us to reason within the metalanguage that any name generated with (new) is different from other names. This definition references 'disjoint coproducts' and

$$[\![\lambda x.M]\!] \stackrel{\text{def}}{=} [\lambda x.[\![M]\!]] \qquad [\![x]\!] \stackrel{\text{def}}{=} [x] \qquad [\![\mathtt{true}]\!] \stackrel{\text{def}}{=} [\mathtt{true}] \qquad [\![\mathtt{false}]\!] \stackrel{\text{def}}{=} [\mathtt{false}]$$

$$[\![M = N]\!] \stackrel{\text{def}}{=} \mathtt{let}\ m \leftarrow [\![M]\!]\ \mathtt{in}\ \mathtt{let}\ n \leftarrow [\![N]\!]\ \mathtt{in}\ [m = n] \quad [\![M\,N]\!] \stackrel{\text{def}}{=} \mathtt{let}\ f \leftarrow [\![M]\!]\ \mathtt{in}\ \mathtt{let}\ x \leftarrow [\![N]\!]\ \mathtt{in}\ f(x)$$

$$[\![\nu x.M]\!] \stackrel{\text{def}}{=} \mathtt{let}\ x \leftarrow \mathtt{new}\ \mathtt{in}\ [\![M]\!] \quad [\![\mathtt{if}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2]\!] \stackrel{\text{def}}{=} \mathtt{let}\ b \leftarrow [\![M]\!]\ \mathtt{in}\ \mathtt{if}\ b\ \mathtt{then}\ [\![N_1]\!]\ \mathtt{else}\ [\![N_2]\!]$$

Fig. 4. Interpretation of ν-calculus expressions in a categorical model, using its metalanguage [Stark 1996, Fig. 5].

'decidable equality', concepts from categorical logic, but we will not assume familiarity with these in the rest of the article except in the proof of Thm 3.8.

Denotational Semantics. In any categorical model we can interpret ν-calculus types (Fig. 2) as objects, using the standard call-by-value translation into the monadic metalanguage: $[\![\mathsf{B}]\!] \stackrel{\text{def}}{=} B$, $[\![\mathsf{N}]\!] \stackrel{\text{def}}{=} N$ and $[\![\sigma \to \tau]\!] \stackrel{\text{def}}{=} [\![\sigma]\!] \to T[\![\tau]\!]$. This is extended to contexts: $[\![\Gamma]\!] \stackrel{\text{def}}{=} \prod_{(x\colon \tau)\in\Gamma} [\![\tau]\!]$. A ν-calculus expression $\Gamma \vdash M : \tau$ is routinely interpreted as a morphism $[\![\Gamma]\!] \to T[\![\tau]\!]$ by induction on the structure of M (Fig. 4).

Using the categorical limits and the equality test on N, we can build a subobject $N^{\neq s} \rightarrowtail N^s$ for all finite sets s, modelling the assumption $(\neq s)$ of distinct names. Formally, $N^{\neq s}$ is the equalizer of $(n : N^s \vdash \bigvee_{i\neq j}(n_i = n_j) : B)$ and $(n : N^s \vdash \mathtt{false} : B)$. For expressions $M \in \mathrm{Exp}_\tau(s)$, we will typically use the restricted interpretation $[\![M]\!]_{\neq s} : N^{\neq s} \rightarrowtail N^s \xrightarrow{[\![M]\!]} T[\![\tau]\!]$.

We note that values $V \in \mathrm{Val}_\tau(s)$ factor through $[-]_{[\![\tau]\!]} : [\![\tau]\!] \to T[\![\tau]\!]$, i.e. we can assume $[\![V]\!] : N^{\neq s} \to [\![\tau]\!]$. Intuitively, the values do not need a top-level monad because they do not generate fresh names.

Any categorical model according to Definition 2.1 is sound and, under mild assumptions, adequate:

Theorem 2.2 ([Stark 1996, Prop. 1–4]). *For any categorical model of the ν-calculus:*

- *The big-step semantics is sound with respect to the denotational semantics: If $s \vdash M \Downarrow_\tau (s')V$ then $[\![M]\!]_{\neq s} = [\![\nu s'.V]\!]_{\neq s}$.*
- *If 1 is not an initial object and $[-]_B : B \to T(B)$ is monic, then the denotational semantics is adequate for observational equivalence: If $[\![M_1]\!]_{\neq s} = [\![M_2]\!]_{\neq s}$ then $M_1 \approx_\tau M_2$, for all expressions $M_1, M_2 \in \mathrm{Exp}_\tau(s)$.*

In Section 6.2 we survey the examples categorical models of the ν-calculus from the literature. In Section 3.3 we show that quasi-Borel spaces form a categorical model.

Categorical models need not identify observationally equivalent terms at higher types. The simplest example of such an equivalence is the privacy equation (3), whose translation into the metalanguage is

$$\mathtt{let}\ a \leftarrow \mathtt{new}\ \mathtt{in}\ [\lambda x.[x = a]] = [\lambda x.[\mathtt{false}]] : T[\![\mathsf{N} \to \mathsf{B}]\!] = T(N \to TB). \tag{10}$$

The metalanguage has extra types such as $(N \Rightarrow B)$ which are not the interpretation of ν-calculus types. So in the metalanguage it is possible to consider the following simpler variation of (10):

$$\mathtt{let}\ a \leftarrow \mathtt{new}\ \mathtt{in}\ [\lambda x.(x = a)] = [\lambda x.\mathtt{false}] : T(N \to B). \tag{PRIV}$$

Note that (PRIV) straightforwardly implies (10) in the metalanguage. So to prove the privacy observational equivalence (3), it is sufficient to find a categorical model that satisfies (PRIV). In Section 4.1 we show that quasi-Borel spaces satisfy (PRIV). We discuss other models in Section 6.2, in particular, neither (PRIV) nor (10) are satisfied in the nominal sets model (14).

We remark that because of the call-by-value semantics of ν-calculus, first-order functions already exhibit an interesting degree of complexity that cannot be simplified by uncurrying. At type $[\![\mathsf{N} \to (\mathsf{N} \to \mathsf{B})]\!] = T(\mathsf{N} \to T(\mathsf{N} \to T\mathsf{B}))$, name-generation effects may occur at three different stages, unlike in the uncurried version $T(\mathsf{N} \times \mathsf{N} \to T\mathsf{B})$.

3 HIGHER-ORDER PROBABILITY

The central new definition of this paper is the random model of the ν-calculus based on quasi-Borel spaces. We recall Borel spaces in Section 3.1, quasi-Borel spaces in Section 3.2, and then explain the model in Section 3.3, in prepration for the full abstraction result in Section 4.

3.1 Rudiments of Measurable Spaces

Probability spaces are traditionally defined in terms of measurable spaces [Kallenberg 2002; Pollard 2001]. A *measurable space* is a set X together with a σ-algebra Σ_X on X. We call a set $U \subseteq X$ *measurable* if $U \in \Sigma_X$. A function $f : X \to Y$ between measurable spaces is *measurable* if for all measurable $A \subseteq Y$, the set $f^{-1}(A)$ is measurable in X.

The measurable spaces and measurable functions form the category **Meas**. This category has products given by equipping $X \times Y$ with the product σ-algebra $\Sigma_X \otimes \Sigma_Y$.

The *Borel σ-algebra* $\Sigma_{\mathbb{R}}$ is the σ-algebra on $\mathbb{R}$ generated by the open intervals. We will always consider $\mathbb{R}$ as a measurable space with the Borel σ-algebra. We say a measurable space X is *discrete* if $\Sigma_X = \mathcal{P}(X)$, where $\mathcal{P}(X)$ denotes the power set of X.

A *measure* on a measurable space X is a σ-additive map $\mu : \Sigma_X \to [0, \infty]$ with $\mu(\emptyset) = 0$. It is *finite* if $\mu(X) < \infty$, *s-finite* if it is the countable sum of finite measures, and a *probability measure* if $\mu(X) = 1$. A *probability space* (X, μ) is a measurable space X and a fixed probability measure μ on X. If μ is a probability measure on X and $f : X \to Y$ is measurable, then the *pushforward measure* $f_*\mu$ on Y is defined by $f_*\mu(U) = \mu(f^{-1}(U))$ for $U \in \Sigma_Y$. If $f : X \to \mathbb{R}$, then we can find the Lebesgue integral $\int_X f(x)\, d\mu(x) \in \mathbb{R}$.

There is a monad $\mathcal{G} : \mathbf{Meas} \to \mathbf{Meas}$ due to [Giry 1982] that assigns to X the space of probability measures $\mathcal{G}X$ on X, with the σ-algebra generated by the maps $\mu \mapsto \mu(U)$ for all $U \in \Sigma_X$. The unit of this monad is the Dirac distribution $X \to \mathcal{G}X, x \mapsto \delta_x$. The bind of this monad consists of the averaging of measures, so that if $f : X \to \mathcal{G}Y$, we get the map $f^* : \mathcal{G}X \to \mathcal{G}Y$ taking $\mu \in \mathcal{G}X$ to the measure $f^*(\mu)(U) = \int_X f(x)(U)\, d\mu(x)$ on Y. In the metalanguage, we can regard $\mathtt{let}\ x \leftarrow \mu\ \mathtt{in}\ f(x)$ $(= f^*(\mu))$ as a generalized integral $\int f(x)\, d\mu(x)$. This monad is strong and commutative (9), which is a categorical way to state Fubini's theorem [Kallenberg 2002, 1.27]. The monad is moreover affine (8), since in general $g(y) = \int g(y)\, d\mu(x)$ for a probability measure μ.

When a probability space (Ω, μ) is fixed, we say a *random variable* A with values in X is a measurable map $A : \Omega \to X$. Two random variables A, B are said to be *equal in distribution*, written $A \overset{d}{=} B$, if they have the same law, i.e. $A_*\mu = B_*\mu$ on X.

The spaces $\mathbb{R}$ and $[0, 1]$ are part of an important class of well-behaved measurable spaces called the standard Borel spaces. A *standard Borel space* is a measurable space that is either countable and discrete or measurably isomorphic to $\mathbb{R}$ with the Borel σ-algebra. Note that this is not the usual definition of standard Borel spaces, which can be found in [Kechris 1987, §12.B] and is equivalent to the one above. In particular, the definition of a standard Borel space ignores any underlying topology.

We refer to measurable subsets of standard Borel spaces as *Borel sets*, measurable maps between standard Borel spaces as *Borel measurable* and denote the full subcategory of standard Borel spaces by **Sbs**.

The standard Borel spaces form a well behaved full subcategory of **Meas** closed under taking countable products and coproducts and the Giry monad. Additionally, Borel subsets of standard Borel spaces are standard Borel [Kechris 1987, §12.B, 13.4, 17.23].

Given a standard Borel space X, we call a probability measure μ on X *atomless* if $\mu(\{x\}) = 0$ for all $x \in X$. We have the following isomorphism theorem for standard Borel spaces with atomless probability measures:

Theorem 3.1 ([Kechris 1987, 17.41])**.** *Let ρ be the uniform measure on $[0,1]$. If X is a standard Borel space and μ an atomless measure on X, then there is a Borel measurable isomorphism $f : [0,1] \to X$ such that $f_*\rho = \mu$.*

Example 3.2. The following are examples of familiar standard Borel spaces with atomless probability measures:

(1) The space $\mathbb{R}$ of real numbers with the Gaussian distribution.
(2) The Cantor space $2^{\mathbb{N}}$, which can be viewed as the space of infinite sequences of coin flips, with the measure generated uniformly on the basic open sets: $\mu(\{s \in 2^{\mathbb{N}} : a \subseteq s\}) = 2^{-|a|}$, where a is a finite sequence of flips.
(3) The circle $\mathbb{T} = [0,1)$ (one-dimensional torus) with the uniform measure.

By Theorem 3.1, these are all isomorphic as probability spaces.

We note that a standard Borel space admitting an atomless probability measure is necessarily uncountable and in bijection with $\mathbb{R}$.

Measurable spaces are satisfactory for first-order probabilistic programming [Kozen 1981; Staton 2017], but a result of Aumann shows that they fail to accommodate higher-order functions.

Theorem 3.3 (Aumann [Aumann 1961])**.** *There is no σ-algebra on the space $2^{\mathbb{R}}$ of measurable functions $\mathbb{R} \to 2$ such that the evaluation map $2^{\mathbb{R}} \times \mathbb{R} \to 2$ is measurable.*

We note that $2^{\mathbb{R}}$ can be identified with the set $\Sigma_{\mathbb{R}}$ of Borel sets in $\mathbb{R}$, and in this case the evaluation map $2^{\mathbb{R}} \times \mathbb{R} \to 2$ is simply the inclusion check $(B, x) \mapsto B \ni x$.

3.2 Preliminaries on Quasi-Borel Spaces

Quasi-Borel spaces [Heunen et al. 2017] are a convenient setting including both measure theory and higher-typed function spaces that are increasingly widely used (e.g. [Lew et al. 2019; Sato et al. 2019; Ścibior et al. 2017; Vandenbroucke and Schrijvers 2020]). They work by first restricting probability theory to the well-behaved domain of standard Borel spaces (§3.1). They then provide a conservative extension to function spaces, achieving cartesian closure. (We survey other models of higher-order probability in Section 6.3.)

Definition 3.4 ([Heunen et al. 2017])**.** A *quasi-Borel space* is a set X together with a collection M_X of distinguished functions $\alpha : \mathbb{R} \to X$ called *random elements*. The collection M_X must satisfy

(1) for every $x \in X$, the constant map $\lambda r.x$ lies in M_X;
(2) if $\alpha \in M_X$ and $\varphi : \mathbb{R} \to \mathbb{R}$ is Borel measurable, then $\alpha \circ \varphi \in M_X$; and
(3) if $\{A_i\}_{i=1}^{\infty}$ is a countable Borel partition of $\mathbb{R}$ and $\alpha_i \in M_X$ are given, then the case-split $\alpha(r) = \alpha_i(r)$ for $r \in A_i$ lies in M_X.

A map $f : X \to Y$ between quasi-Borel spaces is a morphism if for all $\alpha \in M_X$ we have $f \circ \alpha \in M_Y$. This defines a category **Qbs**.

We consider the reals with a canonical quasi-Borel structure $M_{\mathbb{R}} = \mathbf{Meas}(\mathbb{R}, \mathbb{R})$. Under that definition, any other quasi-Borel space X satisfies $M_X = \mathbf{Qbs}(\mathbb{R}, X)$. Similarly, we obtain a quasi-Borel structure on the space of booleans by taking $M_2 = \mathbf{Meas}(\mathbb{R}, 2)$ where 2 is the two-point standard Borel space. This has the structure of a coproduct $2 \cong 1 + 1$.

The category **Qbs** is cartesian closed, and we have $Y^X = \mathbf{Qbs}(X, Y)$. By cartesian closure, a map $\mathbb{R} \to Y^X$ is a random element iff its uncurrying $\mathbb{R} \times X \to Y$ is a morphism. For example, $2^{\mathbb{R}}$ comprises the characteristic functions of Borel subsets of $\mathbb{R}$, and the random elements $\mathbb{R} \to 2^{\mathbb{R}}$ are the curried characteristic functions of Borel subsets of $\mathbb{R}^2$.

Any quasi-Borel space (X, M_X) can be equipped with a σ-algebra $\Sigma_{M_X} = \mathbf{Qbs}(X, 2)$, where we identify subsets with their characteristic functions; equivalently, Σ_{M_X} is the greatest σ-algebra making the random elements measurable.

We now define probability theory in this new setting. Given a probability measure $\mu \in \mathcal{G}(\mathbb{R})$ and $\alpha \in M_X$, we can push forward the randomness from $\mathbb{R}$ onto X, obtaining a distribution on X. The definition of the induced σ-algebra Σ_{M_X} makes sure this pushforward is well-defined.

Definition 3.5 ([Heunen et al. 2017])**.** A *probability distribution* on a quasi-Borel space X is an equivalence class $[\alpha, \mu]_\sim$, where $\alpha \in M_X$, $\mu \in \mathcal{G}(\mathbb{R})$ and $(\alpha, \mu) \sim (\alpha', \mu')$ if $\alpha_* \mu = \alpha'_* \mu' \in \mathcal{G}(X, \Sigma_{M_X})$.

We note that the significance of the induced σ-algebra on a quasi-Borel space X is to give a notion of *equality of distributions* on X, which is simply extensional equality of the pushforward measures.

There is a Giry-like strong monad P on **Qbs** which sends X to the space $P(X)$ of probability distributions on X, endowed with the quasi-Borel structure

$$M_{P(X)} = \{\beta : \mathbb{R} \to P(X) \mid \exists \alpha \in M_X, g : \mathbb{R} \to \mathcal{G}\mathbb{R} \text{ measurable s.t. } \beta(r) = [\alpha, g(r)]_\sim\}.$$

For $x \in X$, one can form the Dirac distribution δ_x on X by taking $\delta_x = [\lambda r.x, \mu]_\sim$ for any $\mu \in \mathcal{G}\mathbb{R}$. This forms the unit of the monad. On the other hand, given $f : X \to P(Y)$ and $[\alpha, \mu]_\sim \in P(X)$, we have $f \circ \alpha \in M_{P(Y)}$ so there is some $\beta \in M_Y$ and $g : \mathbb{R} \to \mathcal{G}\mathbb{R}$ such that $f \circ \alpha(r) = [\beta, g(r)]_\sim$. We define a measure on Y by taking $f^*([\alpha, \mu]_\sim) = [\beta, g^*(\mu)]_\sim$. This forms the bind of the monad.

Finally, we note that all of probability theory over standard Borel spaces is the same whether done in **Meas** or **Qbs**.

Proposition 3.6 (Conservativity [Heunen et al. 2017, Prop. 19, 22])**.** *Any measurable space (X, Σ_X) can be regarded as a quasi-Borel space (X, M_{Σ_X}), where $M_{\Sigma_X} = \mathbf{Meas}(\mathbb{R}, X)$. This restricts to a full and faithful embedding* **Sbs** $\to$ **Qbs** *of standard Borel spaces into quasi-Borel spaces that preserves countable products, coproducts and the probability monad.*

Due to this we will identify the standard Borel spaces in both **Meas** and **Qbs** and write say 2 or $\mathbb{R}$ for the quasi-Borel space and measurable space alike.

Probability theory in **Qbs** departs from the traditional foundations only if we go beyond standard Borel spaces. To emphasise this, we briefly make a digression to recall the categorical relationship between quasi-Borel spaces and measurable spaces.

Proposition 3.7 ([Heunen et al. 2017, Prop. 15])**.** *The maps $\Sigma : (X, M_X) \mapsto (X, \Sigma_{M_X})$ and $M : (X, \Sigma_X) \mapsto (X, M_{\Sigma_X})$ are functorial and form an adjunction*

$$\mathbf{Qbs} \underset{M}{\overset{\Sigma}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathbf{Meas}$$

Now consider the quasi-Borel space $2^{\mathbb{R}}$. Using this adjunction (Prop. 3.7), we obtain a σ-algebra $\Sigma_{2^{\mathbb{R}}}$ on the set $2^{\mathbb{R}}$ and a measurable evaluation map $\Sigma(2^{\mathbb{R}} \times \mathbb{R}) \to 2$. We note that this does not contradict Theorem 3.3 because Σ does not preserve products, and indeed the σ-algebra $\Sigma_{2^{\mathbb{R}} \times \mathbb{R}}$ induced from the quasi-Borel space $2^{\mathbb{R}} \times \mathbb{R}$ is strictly larger than the product algebra $\Sigma_{2^{\mathbb{R}}} \otimes \Sigma_{\mathbb{R}}$ (cf. Theorem 4.1, Proposition 5.8, and Observation 5.10).

3.3 Probabilistic Semantics for the ν-Calculus

We can now give probabilistic semantics to the ν-calculus (cf. Def. 2.1) by interpreting names as elements of a probability space and name generation as random sampling.

Theorem 3.8. **Qbs** *is a categorical model of the ν-calculus under the following assignment:*

(1) *the object of names is $N = \mathbb{R}$, and the object of Booleans is $B = 2$;*
(2) *the name-generation monad is $T = P$; and*
(3) new *is given by the Gaussian distribution $\nu \in P(\mathbb{R})$.*

Moreover, it is adequate: If $[\![M_1]\!]_{\neq s} = [\![M_2]\!]_{\neq s}$ then $M_1 \approx_\tau M_2$, for all expressions $M_1, M_2 \in \mathrm{Exp}_\tau(s)$.

Proof. Quasi-Borel spaces have the required categorical structure, and the equality test is a Borel measurable map $(=) : \mathbb{R}^2 \to 2$, hence a morphism. The probability monad is commutative (9), i.e. Fubini holds [Heunen et al. 2017, Prop. 22], and affine because $P(1) \cong 1$, i.e. probability measures must have total mass 1. The freshness requirement is the following identity in the internal language of **Qbs**, which reduces by Conservativity (Prop. 3.6) to a statement about ordinary measure theory:

$$x : \mathbb{R} \vdash \texttt{let}\ y \leftarrow \nu\ \texttt{in}\ [(y, y = x)] = \texttt{let}\ y \leftarrow \nu\ \texttt{in}\ [(y, \texttt{false})] : P(\mathbb{R} \times 2)$$

Because ν is atomless, both sides denote the same distribution $\nu \otimes [\texttt{false}]$.

For adequacy, we verify the assumptions of Thm. 2.2. It is clear that $0 \ncong 1$. To see that the unit $[-]_B : B \to P(B)$ at B is monic, notice that by conservativity (Prop 3.6) it is equivalent to check that $2 \to \mathcal{G}(2)$ is injective in ordinary measure theory, which is trivial. □

Remark 3.9. Any choice of standard Borel space and atomless measure will provide us with a model of the ν-calculus. For example, we could consider $2^{\mathbb{N}}$ or $\mathbb{T} = [0, 1)$ with the uniform measure (cf. Example 3.2), or $\mathbb{R}$ with any other atomless distribution.

By Theorem 3.1, all such choices give isomorphic models of the ν-calculus. More specifically, as the choice of standard Borel space and atomless measure completely determine the semantics of the ν-calculus in **Qbs**, we always obtain the same equational theory of the ν-calculus.

We may therefore choose to use any such space and measure when reasoning about the ν-calculus in **Qbs**. We will take advantage of this in Section 4.3, where we will find it convenient to work with the circle $\mathbb{T}$.

By the general properties of categorical models, **Qbs** semantics are sound and adequate for the ν-calculus. In Section 4 we turn to studying the probabilistic semantics at higher types.

Aside on the 'MONO' Requirement. When working with a monadic metalanguage, several authors [Moggi 1991; Stark 1996] ask that a monad T satisfies the requirement

$$[-]_X : X \to TX \text{ is monic for all } X. \tag{MONO}$$

As we now explain, by using 'separated' quasi-Borel spaces we can support the full (MONO) requirement. We mention this for completeness with respect to the literature, and will not use this notion later in this paper. In Stark's adequacy result (Thm. 2.2(2)), he only requires that $[-]_B : B \to T(B)$ be monic (for $B = 1 + 1$).

Definition 3.10. A quasi-Borel space (X, M_X) is *separated* if the maps $X \to 2$ separate points, meaning that for all $x \neq x' \in X$ there is some morphism $f : X \to 2$ such that $f(x) \neq f(x')$.

This is equivalent to saying that the induced σ-algebra Σ_{M_X} on X separates points.

Proposition 3.11. *A quasi-Borel space X is separated if and only if it satisfies the (MONO) rule: the unit $X \to P(X)$ of the probability monad is injective.*

Additionally, we have: standard Borel spaces are separated; if X, Y are separated, so is $X \times Y$; if Y is separated, so is Y^X; and for every X, $P(X)$ is separated.

Proof notes. The first part follows because for $f : X \to 2$ and $x \in X$, we have $\int_X f(y)\, d\delta_x(y) = f(x)$. The rest is routine calculation. □

Therefore we could model the full (MONO) requirement by restricting to *separated* quasi-Borel spaces. Moreover, this would not change the semantic interpretation.

4 FULL ABSTRACTION

In this section, we will prove that **Qbs** is a fully abstract model of the ν-calculus at first-order types. This will proceed in three steps, as described in §1.3. We will first prove that privacy holds in **Qbs** (§4.1). We will then construct a normal form invariant observational equivalence at first-order types, eliminating the use of private names (§4.2). Finally, we will make use of a measure-invariant group structure on the set of names and the privacy equation established in §4.1 to prove that **Qbs** validates our normalization and is therefore fully abstract at first-order types (§4.3).

4.1 The Privacy Equation

Theorem 4.1 (Privacy for **Qbs**)**.** **Qbs** *satisfies* (PRIV)*. This means that the random singleton is indistinguishable from the empty set:*

$$\texttt{let}\ a \leftarrow \nu\ \texttt{in}\ [\{a\}] = [\emptyset] : P(2^{\mathbb{R}}).$$

In particular, **Qbs** *validates the privacy equation* (3)*.*

In statistical notation, we would consider a Borel set-valued random variable $\{X\}$ where $X \sim \nu$. Privacy states that $\{X\} \overset{d}{=} \emptyset$ in distribution. Before presenting the proof, let us consider some examples of measurable operations which we can apply to Borel sets and see why they fail to distinguish $\{X\}$ from $\emptyset$.

Example 4.2. For any fixed number $x_0 \in \mathbb{R}$, the evaluation map $x_0 \in (-) : 2^{\mathbb{R}} \to 2$ is a morphism. However, testing membership of x_0 will almost surely not distinguish $\{X\}$ and $\emptyset$, as X is sampled from an atomless distribution, so

$$\Pr(x_0 \in \{X\}) = \Pr(X = x_0) = 0 = \Pr(x_0 \in \emptyset).$$

This is merely stating freshness: a freshly generated name is distinct from any fixed existing name. As discussed in Eq. (7), this is a strictly weaker statement than privacy, because λ and ν don't commute.

Example 4.3. Example 4.2 shows that Dirac distributions cannot distinguish the random singleton from the empty set. More generally, they cannot be distinguished by s-finite measures. Evaluating an s-finite measure μ is a morphism $2^{\mathbb{R}} \to [0, \infty]$ [Ścibior et al. 2017, §4.3]. However because the set of atoms of μ is countable, we have $\mu(\{X\}) = 0 = \mu(\emptyset)$ almost surely.

Example 4.4. In Section 1.2 we discussed the Boolean existence function (6), recalling that if it was in a model then the privacy equation (3) would not hold. As we suggested, this function is incompatible with Borel-based probability. We can now be precise: the nonemptiness check $\exists : 2^{\mathbb{R}} \to 2$ is not a quasi-Borel morphism.

To see that this is the case, recall that there exists a Borel subset $B \subseteq \mathbb{R}^2$ of the plane whose projection $\pi(B)$ is not Borel [Kechris 1987, 14.2]. The characteristic function $\chi_B : \mathbb{R} \times \mathbb{R} \to 2$ is a morphism, and so is its currying $\beta : \mathbb{R} \to 2^{\mathbb{R}}$. However, the characteristic function $\chi_{\pi(B)} = \exists \circ \beta$ is

not a morphism because $\pi(B)$ is not measurable. Therefore $\exists : 2^{\mathbb{R}} \to 2$ cannot be a quasi-Borel map.

This implies that the singleton $\{\emptyset\} \subseteq 2^{\mathbb{R}}$ is not measurable. Furthermore, the equality check between sets $2^{\mathbb{R}} \times 2^{\mathbb{R}} \to 2$ is not a morphism in **Qbs**.

As Theorem 4.1 is a statement about measures on $2^{\mathbb{R}}$, we must analyze the σ-algebra $\Sigma_{2^{\mathbb{R}}}$ on $2^{\mathbb{R}}$ induced by its quasi-Borel structure.

Notation 4.5. Let $B \subseteq X \times Y$ and $x \in X$. We let $B_x = \{y \in Y \mid (x, y) \in B\}$ denote the vertical section of B at x.

Recall that we can identify the space $2^{\mathbb{R}} = \mathbf{Qbs}(\mathbb{R}, 2)$ with the Borel subsets of $\mathbb{R}$. We can similarly identify the set $\mathbf{Qbs}(\mathbb{R} \times \mathbb{R}, 2)$ with the Borel subsets of $\mathbb{R} \times \mathbb{R}$, and by currying this means that the maps in $\mathbf{Qbs}(\mathbb{R}, 2^{\mathbb{R}})$ are exactly the maps $\lambda r.B_r$ for Borel $B \subseteq \mathbb{R} \times \mathbb{R}$. If $B \subseteq \mathbb{R} \times \mathbb{R}$ and $\mathcal{U} \subseteq 2^{\mathbb{R}}$, we note that

$$(\lambda r.B_r)^{-1}(\mathcal{U}) = \{r \in \mathbb{R} \mid B_r \in \mathcal{U}\}.$$

Definition 4.6 ([Kechris 1987])**.** A collection $\mathcal{U} \subseteq 2^{\mathbb{R}}$ of Borel sets is *Borel on Borel* if for all Borel $B \subseteq \mathbb{R} \times \mathbb{R}$, the set $\{r \in \mathbb{R} \mid B_r \in \mathcal{U}\}$ is Borel.

It follows that the σ-algebra $\Sigma_{2^{\mathbb{R}}}$ on $2^{\mathbb{R}}$ induced by the quasi-Borel structure is exactly the collection of Borel on Borel sets. Examples of such families include the family of null sets with respect to a Borel probability measure (Example 4.3) and the family of meager sets [Kechris 1987, §18.B].

Definition 4.7 ([Kechris 1987])**.** Let X be a standard Borel space. Two disjoint sets $A, A' \subseteq X$ are said to be *Borel separable* if there is a Borel set $B \subseteq X$ such that $A \subseteq B$ and $A' \cap B = \emptyset$. A, A' are *Borel inseparable* if no such set exists.

Theorem 4.8 (Becker [Kechris 1987, 35.2])**.** *There exists a Borel set $B \subseteq \mathbb{R} \times \mathbb{R}$ such that the sets*

$$B^0 = \{x \in \mathbb{R} \mid B_x = \emptyset\} \quad \textit{and} \quad B^1 = \{x \in \mathbb{R} \mid B_x \textit{ is a singleton}\}$$

are Borel inseparable.

Using this, we prove that quasi-Borel spaces validate privacy.

Lemma 4.9. *Let $\mathcal{U} \subseteq 2^{\mathbb{R}}$ be Borel on Borel. If $\emptyset \in \mathcal{U}$ then $\{r\} \in \mathcal{U}$ for all but countably many $r \in \mathbb{R}$.*

Proof. Let $A = \{r \in \mathbb{R} \mid \{r\} \notin \mathcal{U}\}$. This is a Borel set because $\mathcal{U}$ is Borel on Borel. Borel subsets of standard Borel spaces are standard Borel, so A is standard Borel.

Now suppose for the sake of contradiction that A were uncountable. Because A is standard Borel it is isomorphic to $\mathbb{R}$. Fixing such an isomorphism, we have by Theorem 4.8 a Borel set $B \subseteq \mathbb{R} \times A$ such that B^0, B^1 are Borel inseparable.

However, if $r \in B^0$ then $B_r = \emptyset \in \mathcal{U}$. On the other hand, if $r \in R^1$ then $B_r = \{a\}$ for some $a \in A$, and so $B_r = \{a\} \notin \mathcal{U}$. It follows that $B^0 \subseteq \{r \in \mathbb{R} \mid B_r \in \mathcal{U}\}$ and $B^1 \subseteq \{r \in \mathbb{R} \mid B_r \notin \mathcal{U}\}$. As $\mathcal{U}$ is Borel on Borel, $\{r \in \mathbb{R} \mid B_r \in \mathcal{U}\}$ provides a Borel separation of B^0, B^1, a contradiction. □

Proof of Theorem 4.1. To show that these two quasi-Borel measures are equal, we must check that the pushforward measures agree on the measurable space $(2^{\mathbb{R}}, \Sigma_{2^{\mathbb{R}}})$, meaning that for $\mathcal{U} \in \Sigma_{2^{\mathbb{R}}}$,

$$\emptyset \in \mathcal{U} \iff \nu\{r \in \mathbb{R} \mid \{r\} \in \mathcal{U}\} = 1.$$

Every such $\mathcal{U}$ is Borel on Borel, and by possibly taking complements we can assume that $\emptyset \in \mathcal{U}$. By Lemma 4.9 the set $\{r \in \mathbb{R} \mid \{r\} \in \mathcal{U}\}$ is co-countable, and because ν is atomless this must have ν-measure 1. □

$$b_1 \, R_{\mathrm{B}}^{\mathrm{val}} \, b_2 \Leftrightarrow b_1 = b_2 \qquad n_1 \, R_{\mathrm{N}}^{\mathrm{val}} \, n_2 \Leftrightarrow n_1 \, R \, n_2$$

$$(\lambda x.M_1) \, R_{\sigma\to\tau}^{\mathrm{val}} \, (\lambda x.M_2) \Leftrightarrow \forall R' \colon s_1' \leftrightharpoons s_2', V_1 \in \mathrm{Val}_\sigma(s_1 \uplus s_1'), V_2 \in \mathrm{Val}_\sigma(s_2 \uplus s_2'), \\ V_1 \, (R \uplus R')_\sigma^{\mathrm{val}} \, V_2 \Rightarrow M_1[V_1/x] \, (R \uplus R')_\tau^{\mathrm{exp}} \, M_2[V_2/x]$$

$$M_1 \, R_\tau^{\mathrm{exp}} \, M_2 \Leftrightarrow \exists R' \colon s_1' \leftrightharpoons s_2', V_1 \in \mathrm{Val}_\sigma(s_1 \uplus s_1'), V_2 \in \mathrm{Val}_\sigma(s_2 \uplus s_2'), \\ s_1 \vdash M_1 \Downarrow_\sigma (s_1')V_1 \,\&\, s_2 \vdash M_2 \Downarrow_\sigma (s_2')V_2 \,\&\, V_1 \, (R \uplus R')_\sigma^{\mathrm{val}} \, V_2$$

Fig. 5. Stark's logical relation

We offer some comments about this proof: the strategy we employed generalizes beyond the category of quasi-Borel spaces. Take any model of higher-order probability which agrees with standard Borel spaces on ground types, that is all morphisms $\mathbb{R} \to 2$ are measurable and all measurable maps $\mathbb{R}^2 \to 2$ are morphisms. Then this Borel on Borel property is a necessary constraint on second-order functions $2^{\mathbb{R}} \to 2$, arising from cartesian closure alone. In this case, Lemma 4.9 applies and it is inconsistent for such morphisms to tell apart the empty set from a random singleton with positive probability.

It is now merely an extensionality aspect of **Qbs** that these constraints are also sufficient, and that the inability to distinguish the empty set from singletons implies equality in distribution. The category of sheaves in [Staton et al. 2016] features a more intensional probability monad, where the two sides of the privacy equation presumably cannot be identified.

4.2 A Normal Form for Privacy

The privacy equation is a crucial stepping stone to full abstraction at first-order types. In Section 4.3 we will show that all other first-order observational equivalences can be reduced to privacy. In order to do this, we will first define a syntactic procedure to eliminate private names. Intuitively, private names are names that are not leaked to the environment — if they are not already known outside the program, then they have no observable effects. In this section, we will provide a concrete definition of private names in terms of a logical relation originally developed in [Pitts and Stark 1993], and we will construct a normal form invariant under observational equivalence that eliminates the use of private names in first-order terms.

Let s_1, s_2 be sets of free names; we write $R \colon s_1 \leftrightharpoons s_2$ for a partial bijection or *span* between s_1 and s_2. We write $R \uplus R'$ for the disjoint union of spans between disjoint sets of names, and we write $\mathrm{id}_s \colon s \uplus t_1 \leftrightharpoons s \uplus t_2$ to denote the partial bijection defined that is the identity on s and undefined on t_1, t_2. Stark [Pitts and Stark 1993] defines two families of relations $R_\tau^{\mathrm{val}} \subseteq \mathrm{Val}_\tau(s_1) \times \mathrm{Val}_\tau(s_2)$ and $R_\tau^{\mathrm{exp}} \subseteq \mathrm{Exp}_\tau(s_1) \times \mathrm{Exp}_\tau(s_2)$ by mutual induction, given in Fig. 5.

We note that R_τ^{val} and R_τ^{exp} coincide at values, so we will simply write the relations as R_τ. Additionally, by renaming related names we can without loss of generality reduce any span R to a subdiagonal, writing $s_i = s \uplus t_i$ and $R = \mathrm{id}_s$.

The logical relation agrees with observational equivalence ($\approx$) at first-order types:

Theorem 4.10 ([Pitts and Stark 1993, Theorem 22])**.** *Let τ be a first-order type. Then for $M_1, M_2 \in \mathrm{Exp}_\tau(s)$ we have*

$$M_1 \approx_\tau M_2 \Leftrightarrow M_1 \, (\mathrm{id}_s)_\tau \, M_2$$

It is important to note that the logical relation is defined at all types τ, but the relation at first-order types need only quantify over smaller first-order or ground types, making it possible to reason about observational equivalence of such terms inductively. In this paper we will primarily focus on the logical relation at first order types. In this setting we can tighten up Theorem 4.10 further, as we will explain: for any $s' \subseteq s$, $(\mathrm{id}_{s'})_\tau$ is a partial equivalence relation whose domain comprises those expressions that donâĂŹt leak any names when s' is public, and $(\mathrm{id}_{s'})_\tau$ relates expressions whose behaviours are equivalent when s' is public.

Example 4.11. The privacy equation for the ν-calculus (3) can be established by means of this logical relation. Because $\{a, x\} \vdash (x = a) \Downarrow_{\mathsf{B}} \mathtt{false}$ whenever a, x are distinct names, the logical relation implies that

$$\lambda x.(x = a)\ (\mathrm{id}_\emptyset)_{\mathsf{N}\to\mathsf{B}}\ \lambda x.\mathtt{false},$$

so that intuitively a is private in $\lambda x.(x = a)$. This in turn implies that

$$\nu a.\lambda x.(x = a)\ (\mathrm{id}_\emptyset)_{\mathsf{N}\to\mathsf{B}}\ \lambda x.\mathtt{false},$$

which by Theorem 4.10 establishes the privacy equation of the ν-calculus.

Example 4.12. For names a, b, let $\lambda x.(a\ b)x : \mathsf{N} \to \mathsf{N}$ denote the term

$$\lambda x.\mathtt{if}\ (x = a)\ \mathtt{then}\ b\ \mathtt{else}\ \mathtt{if}\ (x = b)\ \mathtt{then}\ a\ \mathtt{else}\ x.$$

This is the transposition of a, b, swapping a and b and otherwise behaving as the identity. It is clear that $\lambda x.(a\ b)x\ (\mathrm{id}_{\{a,b\}})_{\mathsf{N}\to\mathsf{N}}\ \lambda x.(a\ b)x$. One can also verify that $\lambda x.(a\ b)x\ (\mathrm{id}_\emptyset)_{\mathsf{N}\to\mathsf{N}}\ \lambda x.(a\ b)x$ as well. Here we no longer allow relations to be made with the names a, b, which we think of as private. Similarly, one can check that $\lambda x.(a\ b)x\ (\mathrm{id}_\emptyset)_{\mathsf{N}\to\mathsf{N}}\ \lambda x.x$, so that

$$\nu a.\nu b.\lambda x.(a\ b)x\ (\mathrm{id}_\emptyset)_{\mathsf{N}\to\mathsf{N}}\ \lambda x.x$$

and by Theorem 4.10 $\nu a.\nu b.\lambda x.(a\ b)x$ is observationally equivalent to the identity.

We note that it is not the case that $\lambda x.(a\ b)x\ (\mathrm{id}_{\{a\}})_{\mathsf{N}\to\mathsf{N}}\ \lambda x.(a\ b)x$, as this would require that $b\ (\mathrm{id}_{\{a\}})\ b$. The same holds if we swap a for b. It is therefore apparent that the logical relations capture some of the connections between names; in this case, that if a or b are known, then by passing them as an argument to $\lambda x.(a\ b)x$ the other will be made public as well.

We notice that in these examples, private names are unmatched by spans. Intuitively, this is because the unmatched names do not affect the (observational) semantics of the terms; if we do not already know what they are, then they have no observable effects. In general, given $M \in \mathrm{Exp}_\tau(s \uplus t)$, we are interested in the names in t with observable effects given that the names in s are known to the environment. This motivates Definition 4.17 of private and leaked names.

Notation 4.13. If $R\colon s_0 \leftrightharpoons s_1$ and $S\colon s_1 \leftrightharpoons s_2$ are spans, we let $R;S$ denote the *composition of relations*, meaning that $m(R;S)n$ if there is some z such that mRz and zSn.

Lemma 4.14. *The logical relations are transitive at first-order types. This means that if σ is a first-order type, $M_i \in \mathrm{Exp}_\sigma(s_i)$ for $i = 0, 1, 2$ and $R\colon s_0 \leftrightharpoons s_1, S\colon s_1 \leftrightharpoons s_2$ are spans such that $M_0\ R_\sigma\ M_1$ and $M_1\ S_\sigma\ M_2$, then $M_0\ (R;S)_\sigma\ M_2$.*

Proof. This follows by induction on the type σ. □

Proposition 4.15. *Let σ be a first-order type and $M \in \mathrm{Exp}_\sigma(s \uplus t)$. There is a least $u \subseteq t$ such that $M\ (\mathrm{id}_{s\uplus u})_\sigma\ M$.*

Proof. If $u_0, u_1 \subseteq t$, $M\ (\mathrm{id}_{s\uplus u_0})_\sigma\ M$ and $M\ (\mathrm{id}_{s\uplus u_1})_\sigma\ M$, then $\mathrm{id}_{s\uplus u_0}; \mathrm{id}_{s\uplus u_1} = \mathrm{id}_{s\uplus(u_0\cap u_1)}$ so by transitivity (4.14) we have $M\ (\mathrm{id}_{s\uplus(u_0\cap u_1)})_\sigma\ M$. We can therefore take u to be the intersection of all such sets. □

Proposition 4.16. *Let σ be a first-order type. Let $M_i \in \mathrm{Exp}_\sigma(s \uplus t_i)$ and suppose there is some $R\colon t_1 \leftrightharpoons t_2$ such that $M_1\ (\mathrm{id}_s \uplus R)_\sigma\ M_2$. Let $u_i \subseteq t_i$ be the least set such that $M_i\ (\mathrm{id}_{s\uplus u_i})_\sigma\ M_i$. Then after possibly renaming names in u_i we have $u_1 = u_2 = u$, $\mathrm{id}_u \subseteq R$ and $M_1\ (\mathrm{id}_{s\uplus u})_\sigma\ M_2$.*

Proof. We know that $R; R^{-1} = \mathrm{id}_{\mathrm{dom}(R)}$, so $M_1\ (\mathrm{id}_{s\uplus dom(R)})_\sigma\ M_1$ by transitivity (4.14). As u_1 is least with this property, $u_1 \subseteq dom(R)$.

Now consider the restriction $R \restriction_{u_1}$ of R to u_1. Because $R \restriction_{u_1} = \mathrm{id}_{u_1}; R$, we have by transitivity that $M_1\ (\mathrm{id}_s \uplus R \restriction_{u_1})_\sigma\ M_2$.

A symmetric argument shows that $R \restriction_{u_1}$ is a bijection of u_1 onto u_2. Therefore, after renaming names, we can assume that $u_1 = u_2 = u$ and $R \restriction_u = \mathrm{id}_u$. □

Definition 4.17 (Private and Leaked Names)**.** Let $M \in \mathrm{Exp}_\tau(s \uplus t)$. We define *the set of leaked names in M relative to s*, denoted by $\mathrm{Leak}(M, s)$, to be the least $u \subseteq t$ such that $M\ (\mathrm{id}_{s\uplus u})_\tau\ M$. We call the names that are not leaked *private relative to s*, denoted $\mathrm{Priv}(M, s) = t \setminus \mathrm{Leak}(M, s)$. Given a type τ and a set of names s, we let

$$\mathrm{Safe}^s_\tau = \{M \in \mathrm{Exp}_\tau(s \uplus t) \mid \mathrm{Leak}(M, s) = \emptyset\} = \{M \in \mathrm{Exp}_\tau(s \uplus t) \mid M\ (\mathrm{id}_s)_\tau\ M\}$$

be the set of terms that leak no names relative to s. If s is empty, we write $\mathrm{Priv}(M)$, $\mathrm{Leak}(M)$ and Safe_τ.

Remark 4.18. By Lemma 4.14 and Proposition 4.16, the relation $(\mathrm{id}_s)_\tau$ induces an equivalence relation on Safe^s_τ. In fact, this corresponds to the usual notion of reducibility by logical relations. Concretely, one could equivalently define Safe^s_τ directly as follows:

$$\mathtt{true}, \mathtt{false} \in \mathrm{Safe}^s_{\mathsf{B}} \qquad n \in \mathrm{Safe}^s_{\mathsf{N}} \Leftrightarrow n \in s$$
$$\lambda x.M \in \mathrm{Safe}^s_{\sigma\to\tau} \Leftrightarrow \forall s', V \in \mathrm{Safe}^{s\uplus s'}_\sigma, M[V/x] \in \mathrm{Safe}^{s\uplus s'}_\tau$$
$$M \in \mathrm{Safe}^s_\tau \Leftrightarrow \exists s', V \in \mathrm{Safe}^{s\uplus s'}_\tau, M \Downarrow (s')V.$$

Example 4.19. We have $\mathrm{Priv}(\lambda x.(x = a)) = \{a\}$ and $\lambda x.(x = a) \in \mathrm{Safe}_{\mathsf{N}\to\mathsf{B}}$ (cf. Example 4.11). Similarly, $\mathrm{Priv}(\lambda x.(a\ b)x) = \{a, b\}$ and $\lambda x.(a\ b)x \in \mathrm{Safe}_{\mathsf{N}\to\mathsf{N}}$ (cf. Example 4.12).

In Examples 4.11, 4.12 and 4.19, we identified private names and found logically related terms that eliminate them. We will now show that this can be done for all terms of first-order type by constructing a normal form that recursively eliminates private names.

Notation 4.20. If $s = \{n_1, \ldots, n_k\}$ is a set of names, we write $\nu s.M$ as shorthand for $\nu n_1.\ldots.\nu n_k.M$. We also write

$$\mathtt{if}\ x = n \in s\ \mathtt{then}\ M_n\ \mathtt{else}\ M_0$$

as shorthand for

$$\mathtt{if}\ x = n_1\ \mathtt{then}\ M_{n_1}\ \mathtt{else}\ \mathtt{if}\ \cdots\ \mathtt{else}\ \mathtt{if}\ x = n_k\ \mathtt{then}\ M_{n_k}\ \mathtt{else}\ M_0.$$

Definition 4.21 (Normal form for privacy)**.** Let σ be a first-order type and let $M \in \mathrm{Safe}^s_\sigma$ for $M \in \mathrm{Exp}_\tau(s \uplus t)$. We define the *normal form* $\langle M, s\rangle$ of M by induction on the type σ as follows:

Ground case: If σ is a ground type and M is a value, then we let $\langle M, s\rangle = M$.

Function case $\mathsf{B} \to \tau$**:** Suppose M is a value of type $\mathsf{B} \to \tau$ and that we have already constructed normal forms for expressions of type τ. Expanding M into its η-normal form, we have

$$M = \lambda x.\mathtt{if}\ x = \mathtt{true}\ \mathtt{then}\ M_{\mathtt{true}}\ \mathtt{else}\ M_{\mathtt{false}}$$

for some $M_{\mathtt{true}}, M_{\mathtt{false}} \in \mathrm{Exp}_\tau(s \uplus t)$. We know $M\ (\mathrm{id}_s)_\sigma\ M$, so we have that $M_{\mathtt{true}}\ (\mathrm{id}_s)_\tau\ M_{\mathtt{true}}$ and $M_{\mathtt{false}}\ (\mathrm{id}_s)_\tau\ M_{\mathtt{false}}$. We then define

$$\langle M, s\rangle = \lambda x.\mathtt{if}\ x = \mathtt{true}\ \mathtt{then}\ \langle M_{\mathtt{true}}, s\rangle\ \mathtt{else}\ \langle M_{\mathtt{false}}, s\rangle.$$

Function case $\mathsf{N} \to \tau$**:** Suppose that M is a value of type $\mathsf{N} \to \tau$ and that we have already constructed normal forms for expressions of type τ. Expanding M to its η-normal form, we have

$$M = \lambda x.\mathtt{if}\ x = n \in s \uplus t\ \mathtt{then}\ M_n\ \mathtt{else}\ M_0$$

for some $M_n \in \mathrm{Exp}_\tau(s \uplus t)$ and $M_0 \in \mathrm{Exp}_\tau(s \uplus t \uplus \{x\})$. In this case, $M\ (\mathrm{id}_s)_\sigma\ M$ implies that $M_0\ (\mathrm{id}_{s\uplus\{x\}})_\tau\ M_0$ and $M_n\ (\mathrm{id}_s)_\tau\ M_n$ for all $n \in s$. We then define

$$\langle M, s\rangle = \lambda x.\mathtt{if}\ x = n \in s\ \mathtt{then}\ \langle M_n, s\rangle\ \mathtt{else}\ \langle M_0, s \uplus \{x\}\rangle.$$

Expression case: Suppose that we have constructed normal forms for values of type σ. Because $M\ (\mathrm{id}_s)_\sigma\ M$, there is some $V \in \mathrm{Val}_\sigma(s \uplus t \uplus w)$ such that $s \uplus t \vdash M \Downarrow_\sigma (w)V$ and $V\ (\mathrm{id}_{s\uplus w'})_\sigma\ V$ for some $w' \subseteq w$. Let $u = \mathrm{Leak}(V, s) \subseteq w'$. Then $V\ (\mathrm{id}_{s\uplus u})_\tau\ V$, so we can define

$$\langle M, s\rangle = \nu u.\langle V, s \uplus u\rangle.$$

If s is empty, we omit it and write $\langle M\rangle$.

Example 4.22. This normal form generalizes the observations in Examples 4.11 and 4.12. Specifically, we have

$$\langle \nu a.\lambda x.(x = a)\rangle = \lambda x.\mathtt{false} \qquad \text{and} \qquad \langle \nu a.\nu b.\lambda x.(a\ b)x\rangle = \lambda x.x.$$

The choice of $u = \mathrm{Leak}(V, s)$ in the expression case of our construction is crucial here; it is of course true that

$$\lambda x.(a\ b)x\ (\mathrm{id}_{\{a,b\}})_{\mathsf{N}\to\mathsf{N}}\ \lambda x.(a\ b)x,$$

but this does not help us identify and eliminate the private names a, b.

Proposition 4.23. *Let τ be a first-order type and $M \in \mathrm{Safe}^s_\tau$. Then*

(1) *if M is a value, so is $\langle M, s\rangle$;*
(2) *the names that appear in $\langle M, s\rangle$ are a subset of the names that appear in M;*
(3) *$\langle M, s\rangle$ eliminates the names in $\mathrm{Priv}(M, s)$ (i.e. $\langle M, s\rangle \in \mathrm{Exp}_\tau(s)$);*
(4) *$\langle M, s\rangle \in \mathrm{Safe}^s_\tau$;*
(5) *$\langle M, s\rangle$ is well-defined up to renaming bound variables and names; and*
(6) *$M\ (\mathrm{id}_s)_\tau\ \langle M, s\rangle$.*

Proof. (1) is clear by construction. (4) follows trivially from (3). We prove (2), (3), (5) and (6) by induction on τ, following the construction of the normal form $\langle M, s\rangle$. For (2) and (3), the induction steps are clear and so is the case where M is a value of type B. If M is a value of type N, then $M\ (\mathrm{id}_s)_\mathsf{N}\ M$ implies that $M \in s$, and so $\langle M, s\rangle = M \in \mathrm{Exp}_\mathsf{N}(s)$. For (5), the cases where M is a value are clear, and the expression case follows because we made a canonical choice of $u = \mathrm{Leak}(V, s)$ in the construction of $\langle M, s\rangle$. For (6), the expression case follows directly from the inductive hypothesis and the definition of logical relations. In the case where M is a value and $\tau = \mathbb{N} \to \sigma$, we η-expand and write

$$M = \lambda x.\mathtt{if}\ x = n \in s \uplus t\ \mathtt{then}\ M_n\ \mathtt{else}\ M_0.$$

We need to verify that $M_0\ (\mathrm{id}_{s\uplus\{x\}})_\sigma\ \langle M_0, s \uplus \{x\}\rangle$ and that $M_n\ (\mathrm{id}_s)_\sigma\ \langle M_n, s\rangle$ for $n \in s$, both of which follow from the inductive hypothesis. The case that M is a value and $\sigma = \mathsf{B} \to \tau$ is handled similarly. □

Example 4.24. As noted in Example 4.19, $\mathrm{Leak}(\lambda x.(x = a)) = \{a\}$ and $\mathrm{Leak}(\lambda x.(a\ b)x) = \{a, b\}$, and these are indeed eliminated from the normal forms computed in Example 4.22.

We can now equate the problem of checking if two terms are observationally equivalent to one of verifying the equality of their normal forms:

Theorem 4.25. *Let σ be a first-order type and let $M_i \in \mathrm{Exp}_\sigma(s \uplus t_i)$ for $i = 1, 2$. The following are equivalent:*

(1) $M_1\ (\mathrm{id}_s)_\sigma\ M_2$;
(2) $M_i \in \mathrm{Safe}^s_\sigma$ *and* $\langle M_1, s\rangle = \langle M_2, s\rangle$ *after possibly renaming bound variables and names.*

Proof. If $M_i \in \mathrm{Safe}^s_\tau$ and $\langle M_1, s\rangle = \langle M_2, s\rangle$, then $\langle M_1, s\rangle\ (\mathrm{id}_s)_\sigma\ \langle M_2, s\rangle$ and so by transitivity of logical relations (4.14) and Proposition 4.23 we have $M_1\ (\mathrm{id}_s)_\sigma\ M_2$.

For the converse, suppose that $M_1\ (\mathrm{id}_s)_\sigma\ M_2$. By transitivity, it is clear that $M_i\ (\mathrm{id}_s)_\sigma\ M_i$.

To show that $\langle M_1, s\rangle = \langle M_2, s\rangle$, we argue by induction, following the construction of the normal forms. The base case is clear. Now consider the inductive step at values. In the case that $\sigma = \mathsf{N} \to \tau$, we η-expand and write

$$M_i = \lambda x.\mathtt{if}\ x = n \in s \uplus t_i\ \mathtt{then}\ M_n^i\ \mathtt{else}\ M_0^i.$$

By definition of logical relations, because $M_1\ (\mathrm{id}_s)_\sigma\ M_2$, we have $M_0^1\ (\mathrm{id}_{s\uplus\{x\}})_\sigma\ M_0^2$ and $M_n^1\ (\mathrm{id}_s)_\sigma\ M_n^2$ for $n \in s$. By our inductive hypothesis, this means that $\langle M_0^1, s \uplus \{x\}\rangle = \langle M_0^2, s \uplus \{x\}\rangle$ and $\langle M_n^1, s\rangle = \langle M_n^2, s\rangle$ for $n \in s$. It follows that $\langle M_1, s\rangle = \langle M_2, s\rangle$. The case that $\sigma = \mathsf{B} \to \tau$ is the same.

In the case of expressions, let $V_i \in \mathrm{Val}_\sigma(s \uplus t_i \uplus t_i')$ be the values such that $s \uplus t_i \vdash M_i \Downarrow (t_i')V_i$. Let $u_i = \mathrm{Leak}(V, s) \subseteq t_i'$. Then $V_i \in \mathrm{Safe}^{s\uplus u_i}_\tau$ and $\langle M_i, s\rangle = \nu u_i.\langle V_i, s \uplus u_i\rangle$. We know that $M_1\ (\mathrm{id}_s)_\sigma\ M_2$, so there is some $R\colon t_1' \leftrightharpoons t_2'$ such that $V_1\ (\mathrm{id}_s \uplus R)_\sigma\ V_2$. By Proposition 4.16, after possibly renaming names we have $u_1 = u_2 = u$, $\mathrm{id}_u \subseteq R$ and $V_1\ (\mathrm{id}_{s\uplus u})_\sigma\ V_2$. We therefore have $\langle V_1, s \uplus u\rangle = \langle V_2, s \uplus u\rangle$ by our inductive hypothesis, and so $\langle M_1, s\rangle = \langle M_2, s\rangle$. □

4.3 Full Abstraction at First-Order Types

At first-order types, it is sufficient to eliminate private names in order to prove abstraction:

Theorem 4.26. *Let $\mathbb{C}$ be a categorical model of the ν-calculus. $\mathbb{C}$ is fully abstract at first-order types if and only if for all first-order types τ and all $M \in \mathrm{Exp}_\tau(s)$, we have*

$$[\![M]\!]_{\neq s} = [\![\langle M, s\rangle]\!]_{\neq s}. \tag{11}$$

Proof. That this is necessary is clear, as by Proposition 4.23 normal forms preserve logical relations and therefore (by Theorem 4.10) observational equivalence. To see that it is sufficient, suppose that $\mathbb{C}$ satisfies (11) and let $M_1, M_2 \in \mathrm{Exp}_\tau(s)$ for a first-order type τ. If $M_1 \approx_\tau M_2$, then by Theorems 4.10 and 4.25 $\langle M_1, s\rangle = \langle M_2, s\rangle$, and so

$$[\![M_1]\!]_{\neq s} = [\![\langle M_1, s\rangle]\!]_{\neq s} = [\![\langle M_2, s\rangle]\!]_{\neq s} = [\![M_2]\!]_{\neq s}.$$ □

For the remainder of this section, we will let the space of names be the circle $\mathbb{T} = [0, 1)$ and we will let ν be the uniform measure on $\mathbb{T}$ (we may assume this is the case by Remark 3.9). We choose to work with the circle as there is a canonical group structure $(\mathbb{T}, +)$ on $\mathbb{T}$, namely addition modulo 1, that is both compatible with the measurable structure (and hence, by Prop. 3.6, the quasi-Borel structure) of $\mathbb{T}$ and is *ν-invariant*. This means that the maps $+\colon \mathbb{T} \times \mathbb{T} \to \mathbb{T}$ and $-\colon \mathbb{T} \times \mathbb{T} \to \mathbb{T}$ are quasi-Borel, and for all $g \in \mathbb{T}$ and $B \subseteq \mathbb{T}$ Borel we have $\nu(g + B) = \nu(B)$. More generally, this implies that for all $f : \mathbb{T} \to P(X)$ and $g \in \mathbb{T}$, we have

$$\mathrm{let}\ x \leftarrow \nu\ \mathrm{in}\ f(g + x) = \int_{\mathbb{T}} f(g + x)d\nu(x) = \int_{\mathbb{T}} f(x)d\nu(x) = \mathrm{let}\ x \leftarrow \nu\ \mathrm{in}\ f(x).$$

The idea of ν-invariance will be used to treat private names as interchangeable in **Qbs**.

We will now use the ν-invariant group structure on $\mathbb{T}$, along with privacy, to prove that passing to normal forms preserves **Qbs** semantics.

Example 4.27. Consider the transposition $\nu a.\nu b.\lambda x.(a\,b)x$. We have seen that

$$\langle \nu a.\nu b.\lambda x.(a\,b)x\rangle = \lambda x.x.$$

We wish to show that their semantics are equal in **Qbs**, i.e. $[\![\nu a.\nu b.\lambda x.(a\,b)x]\!] = [\![\lambda x.x]\!] : P(P(\mathbb{T})^{\mathbb{T}})$. To do this, we define a function $f : 2^{\mathbb{T}} \times \mathbb{T}^3 \to \mathbb{T}$ as follows:

$$f(B,a,b,x) = \begin{cases} (x-a)+b & \text{if } x-a \in B, \\ (x-b)+a & \text{else if } x-b \in B, \\ x & \text{otherwise.} \end{cases}$$

This function behaves like a generalized transposition, parameterized by a new set-argument B. If $B = \emptyset$, then $f(\emptyset,a,b,x) = x$ is just the identity on x. If $B = \{g\}$ is a singleton, then

$$f(\{g\},a,b,x) = \begin{cases} g+b & \text{if } x = g+a, \\ g+a & \text{else if } x = g+b, \\ x & \text{otherwise,} \end{cases}$$

so that f is a transposition whose parameters have been shifted by g.

We then take $f' : 2^{\mathbb{T}} \times \mathbb{T}^2 \to P(P(\mathbb{T})^{\mathbb{T}})$ to be the map $f'(B,a,b) = [\lambda x.[f(B,a,b,x)]]$, so that

$$f'(\emptyset,a,b) = [\![\lambda x.x]\!] \quad \text{and} \quad f'(\{g\},a,b) = [\![\lambda x.(a\,b)x]\!](g+a,g+b),$$

and we define $h : 2^{\mathbb{T}} \to P(P(\mathbb{T})^{\mathbb{T}})$ to be

$$h(B) = \mathtt{let}\ a \leftarrow \nu\ \mathtt{in}\ \mathtt{let}\ b \leftarrow \nu\ \mathtt{in}\ f'(B,a,b).$$

It is clear that $h(\emptyset) = [\![\lambda x.x]\!]$. On the other hand, by the ν-invariance of the action we have

$$\begin{aligned} h(\{g\}) &= \mathtt{let}\ a \leftarrow \nu\ \mathtt{in}\ \mathtt{let}\ b \leftarrow \nu\ \mathtt{in}\ [\![\lambda x.(a\,b)x]\!](g+a,g+b) \\ &= \mathtt{let}\ a \leftarrow \nu\ \mathtt{in}\ \mathtt{let}\ b \leftarrow \nu\ \mathtt{in}\ [\![\lambda x.(a\,b)x]\!])(a,b) \\ &= [\![\nu a.\nu b.\lambda x.(a\,b)x]\!], \end{aligned}$$

independently of $g \in \mathbb{T}$.

Our problem now reduces to the privacy equation. Specifically, we have

$$\begin{aligned} [\![\lambda x.x]\!] &= \mathtt{let}\ B \leftarrow [\emptyset]\ \mathtt{in}\ h(B) \\ &= \mathtt{let}\ B \leftarrow (\mathtt{let}\ n \leftarrow \nu\ \mathtt{in}\ [\{n\}])\ \mathtt{in}\ h(B) \\ &= \mathtt{let}\ n \leftarrow \nu\ \mathtt{in}\ h(\{n\}) \\ &= \mathtt{let}\ n \leftarrow \nu\ \mathtt{in}\ [\![\nu a.\nu b.\lambda x.(a\,b)x]\!] \\ &= [\![\nu a.\nu b.\lambda x.(a\,b)x]\!], \end{aligned}$$

where the second equality is (PRIV) and the final equality follows by discardability (8).

Notation 4.28. If $\vec{t} = (t_1,\ldots,t_n)$ is a vector in $\mathbb{T}^n$ and $g \in \mathbb{T}$, we write $g + \vec{t} = (g+t_1,\ldots,g+t_n)$. Additionally, we write $\mathtt{let}\ t \leftarrow \nu$ to be shorthand for drawing t samples in a sequence:

$$\mathtt{let}\ t_1 \leftarrow \nu\ \mathtt{in}\ \cdots \mathtt{let}\ t_k \leftarrow \nu.$$

Now suppose that τ is a first-order type and $M \in \mathrm{Exp}_\tau(s)$. We will prove that $[\![M]\!] = [\![\langle M,s\rangle]\!]$ by constructing a function $f : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s} \to P([\![\tau]\!])$ satisfying

$$f(\emptyset,-) = [\![\langle M,s\rangle]\!]_{\neq s}(-) \text{ and } f(\{n\},-) = [\![M]\!]_{\neq s}(-),$$

as we did in Example 4.27, and applying the privacy equation (PRIV).

We will construct this f inductively, parallel to the construction of the normal forms. In order to do this, we will provide a more general, parametrized version of this construction: given $M \in \mathrm{Safe}^s_\tau$ with names in $s \uplus t$, we will construct a function $f : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s \uplus t} \to P(\llbracket \tau \rrbracket)$ such that

$$f(\emptyset, -, \vec{t}) = \llbracket \langle M, s \rangle \rrbracket_{\neq s}(-) \text{ and } f(\{n\}, -, \vec{t}) = \llbracket M \rrbracket_{\neq s}(-, n + \vec{t}).$$

We will use this parametrized version in the inductive step of our proof.

The construction of f itself is somewhat high-level, and is analogous to the difference between the η-normal form of a term and its normal form. It takes as arguments a set of name-permutations B, a sequence s of potentially leaked names, and a sequence of names t that are guaranteed to remain private. It then identifies the redundant parts of the η-normal form — where we compare against a private name t_i — and instead checks whether the name matches one of the names in $B + t_i$.

By selecting B to be a fresh permutation $\vec{t} \leftrightarrow \vec{t'}$, we recover the semantics of the η-normal form. On the other hand, by letting B be the empty set we skip redundant comparisons against private names, recovering the semantics of the normal form. We can then use the privacy equation to equate these two denotations, proving that each term is denotationally equivalent to its normal form.

Proposition 4.29. *Let τ be a first-order type and let $M \in \mathrm{Exp}_\tau(s \uplus t)$. If $M \in \mathrm{Safe}^s_\tau$, then there is a quasi-Borel map*

$$f : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s \uplus t} \to P(\llbracket \tau \rrbracket)$$

such that

$$f(\emptyset, \vec{s}, \vec{t}) = \llbracket \langle M, s \rangle \rrbracket_{\neq s}(\vec{s}) \text{ and } f(\{g\}, \vec{s}, \vec{t}) = \llbracket M \rrbracket_{\neq s \uplus t}(\vec{s}, g + \vec{t})$$

whenever $(\vec{s}, g + \vec{t}) \in \mathbb{T}^{\neq s \uplus t}$.

In the case that $M = V$ is a value, f factors through the unit of the monad.

Proof. We construct f inductively, in parallel to the construction of the normal forms.

Ground case: If τ is a ground type and V is a value, then $\langle V, s \rangle = V$ so we simply let

$$f(B, \vec{s}, \vec{t}) = \llbracket \langle V, s \rangle \rrbracket(\vec{s}) = [|\langle V, s \rangle|](\vec{s}).$$

Function case $\mathsf{B} \to \tau$: Suppose that V is a value of type $\mathsf{B} \to \tau$ and that we have already constructed these functions for expressions of type τ. We η-expand V, so that

$$V = \lambda x.\mathtt{if}\ x = \mathtt{true}\ \mathtt{then}\ M_{\mathtt{true}}\ \mathtt{else}\ M_{\mathtt{false}}.$$

By definition of logical relations and the normal form we have $M_{\mathtt{true}}, M_{\mathtt{false}} \in \mathrm{Safe}^s_\tau$ and

$$\langle V, s \rangle = \lambda x.\mathtt{if}\ x = \mathtt{true}\ \mathtt{then}\ \langle M_{\mathtt{true}}, s \rangle\ \mathtt{else}\ \langle M_{\mathtt{false}}, s \rangle.$$

By assumption we have functions $f_{\mathtt{true}}, f_{\mathtt{false}} : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s \uplus t} \to P(\llbracket \tau \rrbracket)$ satisfying the conditions of Proposition 4.29 for $M_{\mathtt{true}}$ and $M_{\mathtt{false}}$. We then define $f : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s \uplus t} \to P(\llbracket \tau \rrbracket)^B$ by

$$f(B, \vec{s}, \vec{t}) = \lambda x. \begin{cases} f_{\mathtt{true}}(B, \vec{s}, \vec{t}) & \text{if } x = \mathtt{true}, \\ f_{\mathtt{false}}(B, \vec{s}, \vec{t}) & \text{otherwise.} \end{cases}$$

It is clear that $f(\emptyset, \vec{s}, \vec{t}) = |\langle V, s \rangle|(\vec{s})$ and $f(\{g\}, \vec{s}, \vec{t}) = |V|(\vec{s}, g + \vec{t})$ when $(\vec{s}, g + \vec{t}) \in \mathbb{T}^{\neq s \uplus t}$, so that $[f]$ satisfies Proposition 4.29 for V.

Function case $\mathsf{N} \to \tau$: Suppose that V is a value of type $\mathsf{N} \to \tau$ and that we have already constructed these functions for expressions of type τ. We η-expand V, so that

$$V = \lambda x.\mathtt{if}\ x = n \in s \uplus t\ \mathtt{then}\ M_n\ \mathtt{else}\ M_0.$$

By definition of logical relations and the normal form we have $M_0 \in \mathrm{Safe}_\tau^{s \uplus \{x\}}$, $M_n \in \mathrm{Safe}_\tau^s$ for $n \in s$ and

$$\langle V, s\rangle = \lambda x.\mathtt{if}\ x = n \in s\ \mathtt{then}\ \langle M_n, s\rangle\ \mathtt{else}\ \langle M_0, s \uplus \{x\}\rangle.$$

By assumption we have functions $f_n : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s \uplus t} \to P(\llbracket\tau\rrbracket)$ for $n \in s$ and $f_0 : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s \uplus t \uplus \{x\}} \to P(\llbracket\tau\rrbracket)$ satisfying the conditions of 4.29 for M_n and M_0. Writing $t = (t_1, \ldots, t_k)$, we define $f : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s \uplus t} \to P(\llbracket\tau\rrbracket)^{\mathbb{T}}$ by

$$f(B, \vec{s}, \vec{t}) = \lambda x. \begin{cases} f_n(B, \vec{s}, \vec{t}) & \text{if } x = n \in \vec{s}, \\ \llbracket M_{t_1}\rrbracket(\vec{s}, (x - t_1) + \vec{t}) & \text{else if } (x - t_1) \in B, \\ \ldots & \\ \llbracket M_{t_k}\rrbracket(\vec{s}, (x - t_k) + \vec{t}) & \text{else if } (x - t_k) \in B, \\ f_0(B, \vec{s}, \vec{t}, x) & \text{otherwise.} \end{cases}$$

If $B = \emptyset$, then

$$f(\emptyset, \vec{s}, \vec{t}) = \lambda x. \begin{cases} \llbracket\langle M_n, s\rangle\rrbracket(\vec{s}) & \text{if } x = n \in \vec{s}, \\ \llbracket\langle M_0, s\rangle\rrbracket(\vec{s}) & \text{otherwise} \end{cases}$$

so that $f(\emptyset, \vec{s}, \vec{t}) = |\langle V, s\rangle|(\vec{s})$. On the other hand, if $B = \{g\}$ is a singleton, then

$$f(\{g\}, \vec{s}, \vec{t}) = \lambda x. \begin{cases} \llbracket M_n\rrbracket(\vec{s}, g + \vec{t}) & \text{if } x = n \in \vec{s}, \\ \llbracket M_{t_1}\rrbracket(\vec{s}, g + \vec{t}) & \text{else if } x = g + t_1, \\ \ldots & \\ \llbracket M_{t_k}\rrbracket(\vec{s}, g + \vec{t}) & \text{else if } x = g + t_k, \\ \llbracket M_0\rrbracket(\vec{s}, g + \vec{t}, x) & \text{otherwise} \end{cases}$$

so that $f(\{g\}, \vec{s}, \vec{t}) = |V|(\vec{s}, g + \vec{t})$ when $(\vec{s}, g + \vec{t}) \in \mathbb{T}^{\neq s \uplus t}$. Thus $[f]$ satisfies Proposition 4.29 for V.

Expression case: Suppose that we have constructed these reductions for values of type τ. We have $M\,(\mathrm{id}_s)_\tau\,M$, so by definition of logical relations and the normal form there is some $V \in \mathrm{Val}_\tau(s \uplus t \uplus u \uplus w)$ such that $s \uplus t \vdash M \Downarrow_\tau (u \uplus w)V$ and $u = \mathrm{Leak}(V, s)$. Therefore $V \in \mathrm{Safe}_\tau^{s \uplus u}$ and

$$\langle M, s\rangle = \nu u.\langle V, s \uplus u\rangle.$$

By assumption, there is a function $f_V : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s \uplus t \uplus u \uplus w} \to P(\llbracket\tau\rrbracket)$ satisfying the conditions of Proposition 4.29 for V and $\langle V, s \uplus u\rangle$. We then define $f : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s \uplus t} \to P(\llbracket\tau\rrbracket)$ by

$$f(B, \vec{s}, \vec{t}) = \mathtt{let}\ u \leftarrow \nu\ \mathtt{in}\ \mathtt{let}\ w \leftarrow \nu\ \mathtt{in}\ f_V(B, \vec{s}, \vec{t}, \vec{u}, \vec{w}).$$

It follows that

$$\begin{aligned} f(\{g\}, \vec{s}, \vec{t}) &= \mathtt{let}\ u \leftarrow \nu\ \mathtt{in}\ \mathtt{let}\ w \leftarrow \nu\ \mathtt{in}\ f_V(\{g\}, \vec{s}, \vec{t}, \vec{u}, \vec{w}) \\ &= \mathtt{let}\ u \leftarrow \nu\ \mathtt{in}\ \mathtt{let}\ w \leftarrow \nu\ \mathtt{in}\ \llbracket V\rrbracket_{\neq s \uplus t \uplus u \uplus w}(\vec{s}, g + \vec{t}, \vec{u}, g + \vec{w}) \\ &= \mathtt{let}\ u \leftarrow \nu\ \mathtt{in}\ \mathtt{let}\ w \leftarrow \nu\ \mathtt{in}\ \llbracket V\rrbracket_{\neq s \uplus t \uplus u \uplus w}(\vec{s}, g + \vec{t}, \vec{u}, \vec{w}) \\ &= \llbracket \nu u.\nu w.V\rrbracket_{\neq s \uplus t}(\vec{s}, g + \vec{t}) \\ &= \llbracket M\rrbracket_{\neq s \uplus t}(\vec{s}, g + \vec{t}) \end{aligned}$$

whenever $(\vec{s}, g + \vec{t}) \in \mathbb{R}^{\neq s \uplus t}$, where the third equality follows by ν-invariance and the last by soundness (Theorem 2.2). Similarly, we verify that $f(\emptyset, \vec{s}, \vec{t}) = \llbracket\langle M, s\rangle\rrbracket_{\neq s}(\vec{s})$ by discardability (8) and soundness. □

We note that this construction is not specific to quasi-Borel spaces; it can be performed completely syntactically in a metalanguage asserting that N carries a ν-invariant group structure.

It follows immediately that passing to normal forms preserves **Qbs** semantics, and therefore that **Qbs** is fully abstract at first-order types:

Theorem 4.30. **Qbs** *is fully abstract at first-order types.*

Proof. By Theorem 4.26 it is enough to show that **Qbs** validates passing to normal forms. Let τ be a first-order type and let $M \in \mathrm{Exp}_\tau(s)$. By Proposition 4.29 there is a quasi-Borel map $f : 2^{\mathbb{T}} \times \mathbb{T}^{\neq s} \to P(\llbracket\tau\rrbracket)$ such that

$$f(\emptyset, \vec{s}) = \llbracket\langle M, s\rangle\rrbracket_{\neq s}(\vec{s}) \quad \text{and} \quad f(\{g\}, \vec{s}) = \llbracket M\rrbracket_{\neq s}(\vec{s}).$$

Currying, we get a map $h : 2^{\mathbb{T}} \to P(\llbracket\tau\rrbracket)^{\mathbb{T}^{\neq s}}$ such that

$$h(\emptyset) = \llbracket\langle M, s\rangle\rrbracket_{\neq s} \quad \text{and} \quad h(\{n\}) = \llbracket M\rrbracket_{\neq s}.$$

It follows that

$$\begin{aligned}\llbracket\langle M, s\rangle\rrbracket_{\neq s} &= \mathtt{let}\ B \leftarrow [\emptyset]\ \mathtt{in}\ h(B) = \mathtt{let}\ B \leftarrow (\mathtt{let}\ n \leftarrow \nu\ \mathtt{in}\ [\{n\}])\ \mathtt{in}\ h(B)\\ &= \mathtt{let}\ n \leftarrow \nu\ \mathtt{in}\ h(\{n\}) = \mathtt{let}\ n \leftarrow \nu\ \mathtt{in}\ \llbracket M\rrbracket_{\neq s} = \llbracket M\rrbracket_{\neq s},\end{aligned}$$

where the second equality is (PRIV) and the final equality follows by discardability (8). □

5 STRUCTURAL CONSEQUENCES

In this section, we highlight some consequences our main result has on the category of quasi-Borel spaces and other models of name generation. The privacy equation makes it impossible in **Qbs** to find certain conditional probabilities, as this would require revealing a private name (Prop. 5.2). This means care is needed for Bayesian inference in a higher-typed situation. We will give a broader context for this result using recent notions from synthetic probability theory, allowing us to consider any model of name generation as a categorical model of probability.

Definition 5.1 ([Fritz 2020, 11.1])**.** Let $\mu \in P(X \times Y)$ be a probability distribution and $\mu_X \in P(X)$ its first marginal. A *conditional distribution* for μ is a morphism $\mu_{|X} : X \to P(Y)$ such that

$$\mu = \mathtt{let}\ x \leftarrow \mu_X\ \mathtt{in}\ \mathtt{let}\ y \leftarrow \mu_{|X}(x)\ \mathtt{in}\ [(x, y)].$$

We will now consider the distribution $\mu \in P(2^{\mathbb{R}} \times \mathbb{R})$

$$\mu = \mathtt{let}\ a \leftarrow \nu\ \mathtt{in}\ [(\{a\}, a)] \tag{12}$$

which returns a closure with private name a, but also leaks the name a in the second component.

Proposition 5.2. *In* **Qbs***, conditionals need not exist at function types.*

Proof. By the privacy equation (PRIV), the first marginal of μ (12) equals

$$\mu_1 = \mathtt{let}\ a \leftarrow \nu\ \mathtt{in}\ [\{a\}] = [\emptyset] \quad : P(2^{\mathbb{R}}).$$

If μ admitted a conditional distribution $\mu_{|1} : 2^{\mathbb{R}} \to P(\mathbb{R})$, we would obtain

$$\mu = \mathtt{let}\ A \leftarrow [\emptyset]\ \mathtt{in}\ \mathtt{let}\ b \leftarrow \mu_{|1}(A)\ \mathtt{in}\ [(A, b)] = \mathtt{let}\ b \leftarrow \mu_{|1}(\emptyset)\ \mathtt{in}\ [(\emptyset, b)] \quad : P(2^{\mathbb{R}} \times \mathbb{R}).$$

This is a contradiction, as the predicate $(\ni) : 2^{\mathbb{R}} \times \mathbb{R} \to 2$ is always true for μ, and always false for the RHS. To condition on μ_1 would mean to reconstruct the value a given only access to the marginal $\{a\}$, which is impossible. □

All conditionals from practical statistics (at ground types like $\mathbb{R}$) are still supported by quasi-Borel spaces. The situation is different at function types, but this is not a coincidental pathology of **Qbs**: Name generation offers a systematic reason why conditioning on function types is inconsistent. To make this precise, we will consider any model of name generation as a categorical model of probability theory, and study conditioning in that context. We show that the privacy equation is inconsistent with an axiom called 'positivity', which is valid in traditional measure-theoretic probability, but not in **Qbs** by our full-abstraction result.

Categorical or synthetic probability theory is the abstract axiomatization of probabilistic systems. Its high-level nature ties it closely to the semantics of probabilistic programming languages: One could argue that such languages are precisely the internal languages of synthetic probability theories, and different axioms appear as admissible program equations (see (13)). The subject has been explored among others by [Fritz 2020; Kock 2011; Ścibior et al. 2017]. Of these approaches, we adopt the language of Markov categories which is increasingly widely used [Fritz 2020; Parzygnat 2020; Patterson 2020; Shiebler 2020].

Definition 5.3 ([Fritz 2020, 2.1])**.** A *Markov category* $\mathbb{C}$ is a symmetric monoidal category in which every object X is equipped with the structure of a commutative comonoid $\mathrm{copy}_X : X \to X \otimes X$, $\mathrm{del}_X : X \to I$ satisfying naturality conditions.

Morphisms in a Markov category capture stochastic computation (Markov kernels); the interchange law of $\otimes$ encodes exchangeability/Fubini, and naturality of del the discardability of such computations. copy allows us to introduce correlations. Morphisms $\mu : I \to X$ are called *distributions* on X. Product distributions are formed by the tensor product, and if $\mu : I \to X \otimes Y$ is a distribution, we can take its marginals $\mu_X = (\mathrm{id}_X \otimes \mathrm{del}_Y) \circ \mu, \mu_Y = (\mathrm{del}_X \otimes \mathrm{id}_Y) \circ \mu$.

An important class of examples are Kleisli categories. If T is a commutative and affine monad on a category $\mathbb{C}$ with finite products, then the Kleisli category $\mathbf{Kl}(T)$ is a Markov category [Fritz 2020, 3.2]. Examples are the categories **Set**, **Meas** and **Qbs**, all equipped with their respective probability monads. We observe that name generation (cf. Def. 2.1) is a synthetic probabilistic effect.

Observation 5.4. *For every categorical model $(\mathbb{C}, T)$ of the ν-calculus, the category $\mathbf{Kl}(T)$ is a Markov category.*

Proof. The monad T is assumed commutative and affine, so we apply [Fritz 2020, 3.2]. □

This makes the probabilistic semantics of this paper conceptually very natural: We have taken a synthetic probabilistic effect and given an interpretation using actual randomness. In what follows, we will explore some of the structural differences between name generation and traditional probability theory. By our full abstraction result, this behaviour will apply to quasi-Borel spaces as well.

We let $\mathbb{C}$ denote a Markov category and recall the following definitions

Definition 5.5 ([Fritz 2020, 10.1])**.** A morphism $f : X \to Y$ is *deterministic* if it commutes with copying:

$$\mathrm{copy}_Y \circ f = (f \otimes f) \circ \mathrm{copy}_X.$$

In the case of Kleisli categories, determinism is equivalent to the following program equation in the metalanguage:

$$x : X \vdash \mathtt{let}\ y \leftarrow f(x)\ \mathtt{in}\ [(y, y)] = \mathtt{let}\ y_1 \leftarrow f(x)\ \mathtt{in}\ \mathtt{let}\ y_2 \leftarrow f(x)\ \mathtt{in}\ [(y_1, y_2)] : T(Y \times Y) \quad (13)$$

Note that any morphism that factors through the unit of the monad is deterministic, but the converse is false in general.

Definition 5.6 ([Fritz 2020, 11.22])**.** A Markov category $\mathbb{C}$ is called *positive* if whenever $f : X \to Y$ and $g : Y \to Z$ are such that $g \circ f$ is deterministic, then

$$(g \otimes \mathrm{id}_Y) \circ \mathrm{copy}_Y \circ f = ((g \circ f) \otimes f) \circ \mathrm{copy}_X.$$

This equation is valid in discrete and measure-theoretic probability by [Fritz 2020, 11.25]. We suggest the reading that "irrelevant intermediate results cannot introduce correlations": On the RHS, the output of f is resampled instead of copied. This blatantly fails in the presence of negative probabilities: There is a monad $D_\pm$ on **Set** assigning to X distributions which sum to 1, but whose weights can be negative. Probabilities thus are allowed to interfere destructively. The Kleisli category of $D_\pm$ is still a valid Markov category, and it is in this positivity axiom that its theory deviates from standard probability [Fritz 2020, 11.27]. A consequence of positivity is this:

Proposition 5.7 (One deterministic marginal)**.** *Let $\mathbb{C}$ be a positive Markov category, and $\mu : I \to X \otimes Y$ be a distribution. If the marginal $\mu_X : I \to X$ is deterministic, then $\mu = \mu_X \otimes \mu_Y$.*

Proof. Let $f = \mu$ and $g : X \otimes Y \to X$ be marginalization. By assumption $g \circ f$ is deterministic. The result is obtained by simple string diagram manipulation from the positivity axiom. □

In **Meas**, nothing can be correlated with a constant: If (X, Y) is a joint distribution and $X \overset{d}{=} x_0$ is deterministic, then Y is independent from X. The privacy equation implies that this does not hold for name generation, analogously to Prop. 5.2.

Proposition 5.8. *Any non-degenerate model of the ν-calculus that verifies* (PRIV) *is non-positive.*

Proof. Consider the distribution $\mu = \texttt{let}\ a \leftarrow \texttt{new}\ \texttt{in}\ [(\{a\}, a)]$. Its first marginal is deterministic, as $\mu_1 = \texttt{let}\ a \leftarrow \texttt{new}\ \texttt{in}\ [\{a\}] = [\emptyset]$ by (PRIV). Yet μ is not the product of its marginals $[\emptyset] \otimes \texttt{new}$, as the map $(\ni) : B^N \times N \to B$ distinguishes the two distributions. This violates Prop. 5.7. □

Corollary 5.9. *The category* **Qbs** *is not positive at function spaces.*

We have thus given a natural example of a non-positive Markov category, and this phenomenon has an intuitive meaning in the context of name generation. Any fixed singleton set $\{a\}$ is manifestly distinguishable from $\emptyset$, but only if we know where to look. By randomizing a, its value is perfectly anonymized and this information is lost, leaving us with the empty set. This is reminiscent of a limited form of destructive interference. Note that probabilities in quasi-Borel spaces remain non-negative.

The concept of non-positivity is useful to connect several structural properties of **Qbs**. Firstly, it explains the non-existence of conditionals and disintegrations in Prop 5.2, as by [Fritz 2020, 11.24] conditionals imply positivity. Secondly, the failure of the functor $\Sigma : \mathbf{Qbs} \to \mathbf{Meas}$ (Prop. 3.7) to preserve products is necessary in order to violate Proposition 5.7, as we observe

Observation 5.10. *Let X, Y be quasi-Borel spaces and $\mu \in P(X \times Y)$ such that $\mu_X = [x]$ for some $x \in X$. If $\Sigma(X \times Y) \cong \Sigma X \times \Sigma Y$, then μ is the product of its marginals.*

Proof. If $X \times Y$ carries a product-σ-algebra, the situation reduces to **Meas**, which is positive. □

Proposition 5.8 thus implies that the product $2^{\mathbb{R}} \times \mathbb{R}$ cannot be preserved. Similar arguments can be constructed for other product spaces like $2^{\mathbb{R}} \times 2^{\mathbb{R}}$. Another structural result on quasi-Borel spaces that follows from the methods of §4 concerns the novel status of function spaces.

Proposition 5.11. *The quasi-Borel space $2^{\mathbb{R}}$ is not isomorphic to $M(\Omega)$ for any measurable space Ω.*

Proof. The adjunction $\Sigma \dashv M$ (Prop. 3.7) is idempotent, hence a quasi-Borel space X lies in the essential image of M if and only if $M_X = M_{\Sigma_X}$. We will show that $M_{2^{\mathbb{R}}}$ is strictly smaller than $M_{\Sigma_{2^{\mathbb{R}}}}$. Let $f : \mathbb{R} \to \mathbb{R}$ be a bijective function that is not measurable, and let $A \subseteq \mathbb{R}^2$ be the graph of f. By [Srivastava 1998, Theorem 4.5.2], A is not Borel and hence the map $\alpha : \mathbb{R} \to 2^{\mathbb{R}}, x \mapsto A_x = \{f(x)\}$ does not lie in $M_{2^{\mathbb{R}}}$. However $\alpha \in M_{\Sigma_{2^{\mathbb{R}}}}$, that is α is a measurable map from $\mathbb{R}$ to $(2^{\mathbb{R}}, \Sigma_{2^{\mathbb{R}}})$. Namely, for every $\mathcal{U} \in \Sigma_{2^{\mathbb{R}}}$, we have $\alpha^{-1}(\mathcal{U}) = \{x : \{f(x)\} \in \mathcal{U}\}$. By Lemma 4.9, the set $S = \{x : \{x\} \in \mathcal{U}\}$ is always countable or cocountable, and so is $\alpha^{-1}(\mathcal{U}) = f^{-1}(S)$ by bijectivity of f. So the preimage is a Borel set as desired. □

6 RELATED WORK AND CONTEXT

6.1 Names in Computer Science and Statistics

Names are important in almost every area of practical computer science. There are two main ways to implement name generation: the first is to have one or more servers that deterministically supply fresh names as requested, and the second is to pick them randomly. This paper has emphasised the surprising effectiveness of the latter approach for programming semantics, in that it provides a model that is fully abstract up to first order, not by construction, but by general properties of the real numbers.

Names might be server names in distributed systems, nonces in cryptography, object names in object oriented programming, gensym in Lisp, or abstract memory locations in heap-based programming. Beyond computer science, names play a vital role in logic and set theory. Since this paper is in the theme of probabilistic programming, we emphasise in particular two ways that names are used in probabilistic programming and statistics, and the way that name generation is already understood in terms of randomness there.

- The Dirichlet process can be used as a method for clustering data points where the number of clusters is unknown. The 'base distribution' of a Dirichlet process allocates a label to each cluster that is discovered. It is common to use an atomless distribution such as a Gaussian for this, so that the labels are in effect fresh names for the clusters. In the Church probabilistic programming language, it is common to actually use Lisp's gensym as the base distribution for the Dirichlet process [Roy et al. 2008].
- A graphon is a measurable function $g\colon [0,1]^2 \to [0,1]$, and determines a countably infinite random graph in the following way: we label nodes in the graph with numbers drawn uniformly from $[0,1]$, and there is an edge between two nodes r, s with probability $g(r,s)$. Thus when building a graph node-by-node, the name of each fresh node is, in effect, a real number [Orbanz and Roy 2015].

While many programming languages support name generation directly or through libraries, we have here focussed on the ν-calculus, which is stripped down so that the relationship between name generation and functions can be investigated. There are many other calculi for names, including $\lambda\nu$, which is a call-by-name analogue of the ν-calculus [Odersky 1994], and the π-calculus, for concurrency [Milner 1999]. Moreover, research on the ν-calculus has led to significant developments in different directions, including memory references (e.g. [Jeffrey and Rathke 1999; Laird 2004; Murawski and Tzevelekos 2016]) and cryptographic protocols (e.g. [Sumii and Pierce 2003]). It may well be informative to pursue quasi-Borel based analyses of these applications in the future.

6.2 Models of the ν-Calculus

Arguably the simplest model of the ν-calculus is a set-theoretic model with a special set N of atoms, where abstractness of the atoms is enforced by an invariance property under permutations of the atoms. This model appears in different equivalent guises, including nominal sets and sheaves

on finite sets of names and injective renamings. In this model, types are interpreted as sets, and expressions are interpreted as equivariant functions; see for instance [Pitts 2013, Ch. 9] or [Stark 1994, §3.7]. In nominal sets, equivariance is used to treat private names as interchangeable, which is reminiscent of the idea of ν-invariance in 4.3.

This simple model of nominal sets is very useful, but on its own it is only fully abstract at ground types [Stark 1996, §5]. The privacy law (PRIV) fails because the Boolean existence function $\exists : (N \to B) \to B$ (6) is a morphism of nominal sets, and so we can distinguish the expressions in (PRIV) via the context

$$\mathtt{let}\ f \leftarrow (-)\ \mathtt{in}\ (\exists f) : B. \tag{14}$$

Nominal sets are a Boolean model of set theory [Pitts 2013, Thm. 2.23], and one would necessarily have this kind of existence function $\exists$ in any Boolean model of set theory. Quasi-Borel spaces do form a kind-of model of set theory (a quasitopos), but it is an intuitionistic one, and there is no Boolean existence function (Example 4.4).

To deal with this incompleteness of nominal sets, Stark [Stark 1994, §4.4] proposed a semantic version of the logical relations that we have recalled in Section 4. This model, based on functors between double categories, is fully abstract at first order, as ours is. Subsequently an alternative logical relations model was proposed by [Zhang and Nowak 2003], by working with logical relations over a functor category that more clearly distinguishes between public and private names. **Qbs** is different in spirit to these models, as it is a general purpose model of probability theory rather than a model purpose-built for full abstraction. A quasi-Borel space can be regarded as an $\mathbb{R}$-indexed logical relation (in the sense of [Plotkin 1973]), but it also has a basic role motivated by probability theory.

One curious aspect is that all of these models of the ν-calculus will provide unusual Markov categories (Observation 5.4), i.e. categorical models of probability theory, even if they do not exhibit any randomness in the familiar sense.

Full Abstraction at Higher Types. None of the set-based models justify the following observational equivalence at second-order [Pitts and Stark 1993, Ex. 4(3)]:

$$\nu a.\nu b.\lambda f.(fa \Leftrightarrow fb) \approx_{(\mathsf{N}\to\mathsf{B})\to\mathsf{B}} \lambda f.\mathtt{true} \tag{15}$$

where $\Leftrightarrow$ denotes the biconditional of booleans. To see that this equation fails in the quasi-Borel space model, notice that there is a **Qbs** morphism $(0>) : \mathbb{R} \to 2$ given by $(0>)(r) = \mathtt{true}$ iff $0 > r$, and so we can temporarily add this as a constant to the ν-calculus and keep the rest of the denotational semantics the same. Then $[\![(\lambda f.\mathtt{true})(0>)]\!] = [\![\mathtt{true}]\!]$, but $[\![(\nu a.\nu b.\lambda f(fa \Leftrightarrow fb))(0>)]\!]$ is different; informally it returns `true` with probability 0.5.

To our knowledge, the only models of (15) to date are game-semantic models [Abramsky et al. 2004; Tzevelekos 2008] and bisimulation models [Benton and Koutavas 2008]. In common with our work, normal forms play an implicit role in those models, but those models are very different from ours at higher types. In the future it may be interesting to impose further invariance properties on quasi-Borel spaces to bridge the gap.

Usage of Models in Practice. One major application of models is in validating observational equivalences that may be used for compiler optimizations. In probabilistic programming, optimizations are performed as part of statistical inference algorithms. For instance, discardability (8) and exchangeability (9) are simple but useful translations in practical inference [Murray and Schön 2018; Nori et al. 2014], and partial evaluation and normalization are used in several systems [chieh Shan and Ramsey 2017; Gehr et al. 2020]. Our work in this paper is primarily foundational, but one application is that, in a higher-order probabilistic language, a statistical inference algorithm could

legitimately simplify using our normalization algorithm (§4.2) or higher-typed equations such as the privacy equation (3).

6.3 Other Models of Higher-Order Probability

In this paper we have focused on quasi-Borel spaces, but recently other models of higher-order probability have been proposed. We contend that there are two essential ingredients for using a model of higher-order probability to model the ν-calculus, with name generation as randomness:

(1) it must support an atomless distribution, such as the normal distribution, on some uncountable space N;
(2) it must support equality checking on that space, as a function $N \times N \to 2$.

Some models, such as probabilistic coherence spaces [Ehrhard et al. 2014], do not seem to support atomless distributions, which makes it unclear how to use them for this purpose. Other models are based on the idea that all functions are continuous or computable, e.g. [Escardo 2009; Huang et al. 2018] and then it is impossible to have equality checking for $N = \mathbb{R}$.

This still leaves several recent models, including the stable cones model [Ehrhard et al. 2018], a function analytic model [Dahlqvist and Kozen 2020], game semantics [Paquet and Winskel 2018], geometry of interaction [Dal Lago and Hoshino 2019], boolean-valued sets [Bacci et al. 2018], a boolean topos model [Simpson 2017], and an operational bisimulation [Lago and Gavazzo 2019]. There are also recent logics for higher order probability [Sato et al. 2019]. We understand from the authors that operational bisimulation violates the privacy law, for an interesting reason, and that the boolean topos model violates it because of booleanness (as above, (14)). It remains to be seen how abstract the other recent models are for interpreting the ν-calculus. We note that [Dahlqvist and Kozen 2020; Ehrhard et al. 2018] are currently focused on call-by-*name* semantics and so it is not obvious how to use them with the call-by-*value* ν-calculus that we considered in this paper (see (7)).

Finally we mention another model of higher-order probability that is purely combinatorial [Staton et al. 2018]. That work emphasizes two views of the same model. From one point of view, the space N is a space of real numbers and supports the beta distributions (which are atomless). From another point of view, N is a space of freshly generated names of urns, and real numbers do not arise. This is not a model of the ν-calculus since it does not support name equality checking, but it is related in spirit nonetheless.

6.4 Beyond ν-Calculus

The ν-calculus describes the basic interaction between functions and name generation. Going further, it is also important to investigate the situation where the names have further meaning or structure. In probabilistic programming and statistics, the reorderability of names amounts to sequence exchangeability (e.g. [Staton et al. 2018]), and this is of fundamental importance in statistics and probabilistic programming. But more elaborate symmetries and exchangeabilities are also important (e.g. [Jung et al. 2020; Orbanz and Roy 2015; Staton et al. 2017]), and we leave this for future work.

ACKNOWLEDGMENTS

We thank Alexander Kechris for a first proof of the privacy equation; we have independently developed a different proof based on Borel inseparability (§4.1). We also thank Ohad Kammar for many insightful comments on an early draft of this work. The work has had three starting points: one in discussions with Alex Simpson in 2013; one in discussions with Cameron Freer and Dan Roy in 2016; and the last following discussions with Ohad Kammar and Prakash Panangaden in 2019. We

also thank Tobias Fritz, Mathieu Huot and Sean Moss for helpful discussions. It has been helpful to present preliminary versions of this work at the LAFI and PPS workshops. This work is supported by EPSRC Grant No. EP/N509711/1, a Royal Society University Research Fellowship, FRQNT Grant No. 290736, NSERC Discovery Grant No. RGPIN-2020-05445, NSERC Discovery Accelerator Supplement No. RGPAS-2020-00097 and NCN Grant Harmonia No. 2018/30/M/ST1/00668.

REFERENCES

S. Abramsky, D. R. Ghica, A. S. Murawski, C.-H. L. Ong, and I. D. B. Stark. 2004. Nominal games and full abstraction for the nu-Calculus. In *Proc. LICS 2004*. 150 – 159.

Robert J. Aumann. 1961. Borel structures for function spaces. *Illinois Journal of Mathematics* 5 (1961).

Giorgio Bacci, Robert Furber, Dexter Kozen, Radu Mardare, Prakash Panangaden, and Dana Scott. 2018. Boolean-valued semantics for stochastic lambda-calculus. In *Proc. LICS 2018*.

Nick Benton and Vasileios Koutavas. 2008. *A Mechanized Bisimulation for the Nu-Calculus.* Technical Report MSR-TR-2008-129. Microsoft Research.

Chung chieh Shan and Norman Ramsey. 2017. Exact Bayesian inference by symbolic disintegration. In *Proc. POPL 2017*.

Fredrik Dahlqvist and Dexter Kozen. 2020. Semantics of higher-order probabilistic programs with conditioning. *Proc. ACM Program. Lang.* 4, POPL, Article 19 (Dec. 2020).

Ugo Dal Lago and Naohiko Hoshino. 2019. The geometry of Bayesian programming. In *Proc. LICS 2019*.

Thomas Ehrhard, Michele Pagani, and Christine Tasson. 2018. Measurable cones and stable, measurable functions. In *Proc. POPL 2018*.

Thomas Ehrhard, Charistine Tasson, and Michele Pagani. 2014. Probabilistic coherence spaces are fully abstract for probabilistic PCF. In *Proc. POPL 2014*. 309–320.

M.H. Escardo. 2009. Semi-decidability of may, must and probabilistic testing in a higher-type setting. In *Proc. MFPS 2009*.

Tobias Fritz. 2020. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Adv. Math.* 370, 107239 (Aug. 2020).

T. Gehr, S. Steffen, and M. T. Vechev. 2020. λPSI: exact inference for higher-order probabilistic programs. In *Proc. PLDI 2020*.

Michèle Giry. 1982. A categorical approach to probability theory. In *Categorical Aspects of Topology and Analysis*. Lecture Notes in Mathematics, Vol. 915. Springer, 68–85.

Chris Heunen, Ohad Kammar, Sam Staton, and Hongseok Yang. 2017. A Convenient Category for Higher-Order Probability Theory. In *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science* (Reykjavík, Iceland) *(LICS '17)*. IEEE Press, Article 77, 12 pages.

Daniel Huang, Greg Morrisett, and Bas Spitters. 2018. An application of computable distributions to the semantics of probabilistic programs. arxiv:1806.07966.

A. Jeffrey and J. Rathke. 1999. Towards a theory of bisimulation for local names. In *Proc. LICS 1999*.

Paul Jung, Jiho Lee, Sam Staton, and Hongseok Yang. 2020. A generalization of hierarchical exchangeability on trees to directed acyclic graphs. *Annales Henri Lebesgue* (2020). to appear.

Olav Kallenberg. 2002. *Foundations of Modern Probability.* Springer, New York.

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

Alexander Kechris. 1987. *Classical Descriptive Set Theory.* Springer.

Anders Kock. 2011. Commutative monads as a theory of distributions. *Theory and Applications of Categories* 26 (Aug. 2011).

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

Ugo Dal Lago and Francesco Gavazzo. 2019. On bisimilarity in lambda calculi with continuous probabilistic choice. *Electron. Notes Theoret. Comput. Sci.* 347 (2019), 121 – 141. Proc. MFPS 2019.

James Laird. 2004. A game semantics of local names and good variables. In *Proc. FOSSACS 2004*. 289–303.

J Lambek and P J Scott. 1988. *Introduction to higher order categorical logic.* CUP.

Alexander K. Lew, Marco F. Cusumano-Towner, Benjamin Sherman, Michael Carbin, and Vikash K. Mansinghka. 2019. Trace types and denotational semantics for sound programmable inference in probabilistic languages. *Proc. ACM Program. Lang.* 4, POPL, Article 19 (Dec. 2019).

Robin Milner. 1999. *Communicating and mobile systems - the Pi-calculus.* CUP.

Eugenio Moggi. 1991. Notions of computation and monads. *Inform. Comput.* 93, 1 (1991), 55 – 92.

Andrzej S. Murawski and Nikos Tzevelekos. 2016. Nominal game semantics. *Found. Trends Program. Lang.* (2016).

Lawrence M. Murray and Thomas B. Schön. 2018. Automated learning with a probabilistic programming language: Birch. *Annual Reviews in Control* 46 (2018), 29 – 43.

Aditya Nori, Chung-Kil Hur, Sriram Rajamani, and Selva Samuel. 2014. R2: An efficient MCMC sampler for probabilistic programs. In *Proc. AAAI 2014*.

Martin Odersky. 1994. A Functional Theory of Local Names. In *Proc. POPL 1994*. 48 – 59.

Peter Orbanz and Daniel M. Roy. 2015. Bayesian models of graphs, arrays and other exchangeable random structures. *IEEE Trans. Pattern Anal. Mach. Intell.* 2 (2015), 437–461.

Hugo Paquet and Glynn Winskel. 2018. Continuous probability distributions in concurrent games. In *Proc. MFPS 2018*. 321–344.

Arthur J. Parzygnat. 2020. Inverses, disintegrations, and Bayesian inversion in quantum Markov categories. arXiv:2001.08375.

Evan Patterson. 2020. *The algebra and machine representation of statistical models.* Ph.D. Dissertation. Stanford University Department of Statistics.

Andrew M. Pitts. 2013. *Nominal Sets: Names and Symmetry in Computer Science.* Cambridge University Press.

Andrew M. Pitts and Ian Stark. 1993. Observable properties of higher order functions that dynamically create local names, or: What's *new*?. In *Proc. MFCS 1993 (Lecture Notes in Computer Science)*. 122–141.

G. D. Plotkin. 1973. *Lambda-definability and logical relations.* Technical Report SAI-RM-4. School of A.I., Univ.of Edinburgh.

David Pollard. 2001. *A users' guide to measure-theoretic probability.* CUP.

Daniel Roy, Vikash Mansinghka, Noah Goodman, and Josh Tenenbaum. 2008. A stochastic programming perspective on nonparametric Bayes. In *Proc. ICML Workshop on Nonparametric Bayes*.

Tetsuya Sato, Alejandro Aguirre, Gilles Barthe, Marco Gaboardi, Deepak Garg, and Justin Hsu. 2019. Formal verification of higher-order probabilistic programs: reasoning about approximation, convergence, bayesian inference, and optimization. *Proc. ACM Program. Lang.* 3, POPL, Article 38 (Jan. 2019), 30 pages.

Adam Ścibior, Ohad Kammar, Matthijs Vákár, Sam Staton, Hongseok Yang, Yufei Cai, Klaus Ostermann, Sean Moss, Chris Heunen, and Zoubin Ghahramani. 2017. Denotational validation of higher-order Bayesian inference. *Proceedings of the ACM on Programming Languages* 2 (Nov. 2017).

Dan Shiebler. 2020. Categorical stochastic processes and likelihood. arXiv:2005.04735.

Alex Simpson. 2017. Probability Sheaves and the Giry Monad. In *Proc. CALCO 2017*.

Shashi M. Srivastava. 1998. *A Course on Borel Sets.* Springer, New York.

Ian Stark. 1994. *Names and Higher-Order Functions.* Ph.D. Dissertation. University of Cambridge. Also available as Technical Report 363, University of Cambridge Computer Laboratory.

Ian Stark. 1996. Categorical models for local names. *LISP and Symbolic Computation* 9, 1 (Feb. 1996), 77–107.

Sam Staton. 2010. Completeness for algebraic theories of local state. In *Proc. FOSSACS 2010*. 48–63.

Sam Staton. 2017. Commutative semantics for probabilistic programming. In *Proc. ESOP 2017*.

Sam Staton, Dario Stein, Hongseok Yang, Nathanael L. Ackerman, Cameron E. Freer, and Daniel M. Roy. 2018. The Beta-Bernoulli process and algebraic effects. *Proc. ICALP 2018*.

S. Staton, H. Yang, N. L.. Ackerman, C. Freer, and D. Roy. 2017. Exchangeable random process and data abstraction. In *Proc. PPS 2017*.

Sam Staton, Hongseok Yang, Frank Wood, Chris Heunen, and Ohad Kammar. 2016. Semantics for probabilistic programming: higher-order functions, continuous distributions, and soft constraints. In *Proc. LICS 2016*. 525 – 534.

Eijiro Sumii and Benjamin C. Pierce. 2003. Logical relations for encryption. *J. Comput. Secur.* 11, 4 (2003), 521–554.

Nikos Tzevelekos. 2008. *Nominal game semantics.* Ph.D. Dissertation. Oxford University Computing Laboratory.

Jan-Willem van de Meent, Brooks Paige, Hongseok Yang, and Frank Wood. 2018. An introduction to probabilistic programming. arxiv:1809.10756.

Alexander Vandenbroucke and Tom Schrijvers. 2020. P$\lambda\omega$NK: functional probabilistic NetKAT. In *Proc. POPL 2020*.

Yu Zhang and David Nowak. 2003. Logical relations for dynamic name creation. In *Proc. CSL 2003*. 575–588.