



HAL
open science

A Domain-Theoretic Approach to Statistical Programming Languages

Jean Goubault-Larrecq, Xiaodong Jia, Clément Théron

► **To cite this version:**

Jean Goubault-Larrecq, Xiaodong Jia, Clément Théron. A Domain-Theoretic Approach to Statistical Programming Languages. *Journal of the ACM (JACM)*, 2023, 10.1145/3611660 . hal-04183840

HAL Id: hal-04183840

<https://inria.hal.science/hal-04183840>

Submitted on 21 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

A Domain-Theoretic Approach to Statistical Programming Languages

[JEAN GOUBAULT-LARRECQ](#), Université Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, France

[XIAODONG JIA](#)^{*}, School of Mathematics & College of Computer Science and Electronic Engineering, Hunan University, China

[CLÉMENT THÉRON](#), Université Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, France

We give a domain-theoretic semantics to a statistical programming language, using the plain old category of `dpos`, in contrast to some more sophisticated recent proposals. Remarkably, our monad of minimal valuations is commutative, which allows for program transformations that permute the order of independent random draws, as one would expect. A similar property is not known for Jones and Plotkin’s monad of continuous valuations. Instead of working with true real numbers, we work with exact real arithmetic, providing a bridge towards possible implementations. (Implementations by themselves are not addressed here.) Rather remarkably, we show that restricting ourselves to minimal valuations does not restrict us much: *all* measures on the real line can be modeled by minimal valuations on the domain \mathbb{R}_\perp of exact real arithmetic. We give three operational semantics for our language, and we show that they are all adequate with respect to the denotational semantics. We also explore quite a few examples in order to demonstrate that our semantics computes exactly as one would expect, and in order to debunk the myth that a semantics based on continuous maps would not be expressive enough to encode measures with non-compact support using only measures with compact support, or to encode measures via non-continuous density functions, for instance. Our examples also include some useful, non-trivial cases of distributions on higher-order objects.

ACM Reference Format:

Jean Goubault-Larrecq, Xiaodong Jia, and Clément Théron. 2023. A Domain-Theoretic Approach to Statistical Programming Languages. 1, 1 (August 2023), 64 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

CONTENTS

Abstract	1
Contents	1
1 Introduction	3
2 Preliminaries, Challenges	5
2.1 Measure theory	5
2.2 Domain theory and topology	6

^{*}Corresponding author.

Authors’ addresses: [Jean Goubault-Larrecq](#), goubault@lsv.fr, Université Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, 4, avenue des sciences, Gif-sur-Yvette, France, 91190; [Xiaodong Jia](#), jia.xiaodong@yahoo.com, School of Mathematics & College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan, China, 410082; [Clément Théron](#), clement.theron@ens-paris-saclay.fr, Université Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, 4, avenue des sciences, Gif-sur-Yvette, France, 91190.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2023 Association for Computing Machinery.

Manuscript submitted to ACM

Manuscript submitted to ACM

1

2.3	Continuous valuations and measures	7
2.4	Is there any trouble with the probabilistic powerdomain?	8
2.5	Scoring, and density functions	8
3	Monads of continuous valuations	9
4	The Fubini-Tonelli theorem, and its unsolved subtle variant on Dcpo	9
5	Minimal valuations, and Fubini-Tonelli again	11
5.1	The monad V_m of minimal valuations	11
5.2	Tensorial strengths	12
5.3	Minimal valuations form a commutative monad on Dcpo , or Fubini-Tonelli again	12
6	Measures on \mathbb{R} , minimal valuations on \mathbb{R}	13
7	The ISPCF Calculus	16
8	The Denotational Semantics of ISPCF	18
9	Examples	20
9.1	Rejection sampling	20
9.2	Generating normal distributions	21
9.3	Generating Lebesgue valuation on \mathbb{R}	23
9.4	Generating exponential distributions	23
9.5	Distributions on higher-order objects	24
10	Operational Semantics	28
10.1	Soundness	31
10.2	Adequacy	32
10.3	Contextual equivalence and the applicative preorder	35
11	A Precise Operational Semantics	36
12	A Sampling-Based Operational Semantics	38
12.1	Random strings	38
12.2	The sampling-based operational semantics	38
12.3	A few notes on implementation	40
13	Related work	41
14	Conclusion	42
	Acknowledgments	42
	References	42
A	Examples continued	45
A.1	The call-by-value versus call-by-name question	45
A.2	Verifying that lebesgue implements Lebesgue valuation on \mathbb{R}	46
A.3	Other implementations of the exponential distribution	47
A.4	Proofs omitted from Section 9.5 (distributions on higher-order objects)	48
A.5	More distributions on higher-order objects	51
B	Other omitted proofs	53
B.1	Proof of Proposition 10.8	53
B.2	Soundness (Section 10.1)	54
B.3	Adequacy (Section 10.2)	58

1 INTRODUCTION

The purpose of this paper is to give an accessible, domain-theoretic semantics to statistical programming languages.

Statistical programming languages such as Church [29], Anglican [85], WebPPL [30] or Venture [57], were introduced as a convenient means to describe and implement stochastic generative processes. Those are randomized programs that describe probability distributions.

Initial proposals focused on implementations. One of the first proposals for a formal semantics of such a language, SFPC, is due to Vákár et al. [81], and is based on *quasi-Borel predomains*, a notion that expands on the clever notion of *quasi-Borel spaces* [38], with additional domain-theoretic structure.

The constructions of [81] are rather involved, and one may wonder there would exist simpler denotational semantics for such languages. We will give one, based on domain theory alone. Domain theory is probably one of the oldest mathematical basis for denotational semantics [70]. The earliest work that considered probabilistic choice and its semantics in domain theory dates back to the pioneering work of Saheb-Djahromi [66] (later corrected in [5]). It is a common belief that domain theory would be inadequate for giving semantics to probabilistic languages. This may be due to a superficial reading of a famous paper by A. Jung and R. Tix [46]. And indeed, there are several purely domain-theoretic semantics of probabilistic languages [32, 33, 44].

However, it is true that statistical probabilistic languages present additional challenges to the semanticist. Those are caused by several additional features that one must take into account: a native type of real numbers, continuous distributions, and perhaps most importantly, *soft constraints* [76]. The latter is a convenient way of implementing the computation of conditional distributions, or smoothed versions therefore, and is implemented by a primitive called **score** in SFPC.

One difference between the quasi-Borel semantics of SFPC [81], or that of PCFSS [10], or that of PPCF [17], with our domain-theoretic semantics does not lie in probabilities or soft constraints, but with the way we handle real numbers. SFPC, PCFSS, and PPCF rely on *true* real numbers, that is, values of type **real** are interpreted as elements of \mathbb{R} . We interpret values of type **real** as *exact* real numbers, namely as elements of a dcpo \mathbb{R}_\perp of interval approximations of real numbers, as in RealPCF and other proposals for exact real arithmetic [8, 15, 19, 20, 39, 55, 58, 64, 78]. This is the most natural choice with domain-theoretic semantics, as \mathbb{R} itself is not a dcpo (more precisely, the topology of \mathbb{R} cannot be obtained as a Scott topology), but embeds naturally in a domain of interval approximations. This is also a natural bridge to possible implementations; although we will not pursue this topic in depth, the final section of this paper will give hints. [Another difference is that our distributions will be continuous valuations, following Jones and Plotkin \[44, 45\], instead of measures; the two notions are very close, though, as we will explain in Section 2.3.](#)

Contributions. Our contributions are as follows.

- First, we give a purely domain-theoretic denotational semantics (Section 8) for a statistical programming language with exact real arithmetic, continuous distributions and soft constraints, featuring full recursion.
- Additionally, our monad of *minimal valuations* implementing probabilistic choice is commutative (Section 5). This is crucial in establishing the correctness of run-of-the-mill program transformations such as permuting the order in which two independent random variables are drawn. Some previous, recent proposals such as [17, 81]

also offer commutative probabilistic choice monad, either based on suitable quotients of spaces of objects that are random variables in spirit in the category of quasi-Borel predomains, or on Markov kernels in a category of cones and stable measurable functions. Our minimal valuations are a simple variant of Jones and Plotkin’s continuous valuations [44, 45], on the category of dcpos and Scott-continuous maps. Obtaining a commutative monad of continuous valuations on dcpos was an open problem until an even more recent paper by Jia (the second author), Lindenhovius, Mislove and Zamdzhiev [42]. The first author made that discovery at the same time that they did (see the final related work section for details). A crucial difference with their work is that we deal with *continuous* distributions. By this, we mean that our semantics (Section 8), and our language ISPCF (Section 7), include such *distributions* as the Lebesgue *valuation* (formally obtained from Lebesgue measure by restriction to the open sets) on $[0, 1]$ or on \mathbb{R} , normal distributions, exponential distributions, and so on. The language of [42] can only handle *discrete* distributions, namely countable weighted sums of Dirac measures; this can be seen, for example, by looking at the statement of their strong adequacy theorem (Theorem 55). Among the less significant differences, we deal with unbounded valuations and not subprobability valuations. That is required in order to give semantics to the **score** primitive, which they do not have (see Section 2.5 for an introduction to scoring).

- We stress that working with minimal valuations instead of general, continuous valuations is a real constraint, or so it seems at first. As we will see, the Lebesgue *valuation* is a continuous valuation on \mathbb{R} that is *not* minimal. In spite of this, and this is perhaps one of the most pleasing aspects of our work, *all* measures on \mathbb{R} (or, more precisely, all continuous valuations obtained from any measure on \mathbb{R} by restriction to the open sets) —including Lebesgue *valuation*—can be realized as minimal valuations on the larger space \mathbb{R}_{\perp} , as we will show in Section 6; and therefore, such distributions are part of the semantics of our language ISPCF.
- We provide an extensive list of examples in Section 9, by which we hope to demonstrate that our approach is capable of defining a rich set of distributions. They also serve to verify that our semantics gives the expected distributions in each case. All our examples are supported with proofs of correctness.
- We also *demonstrate the versatility of our approach in a number of ways*. For example, *and although our language can only express continuous functions*, we *can* define distributions with non-compact support from distributions on *the compact interval* $[0, 1]$ (Section 9.2 and 9.3). Or, we can define distributions from other distributions using density maps that are not continuous; it suffices that their points of discontinuity form a null set (Section 9.4).
- We also give a few examples of constructions of distributions on higher-order objects: for example, the countably infinite product of Lebesgue *valuations* on $[0, 1]$, described as a distribution over values of type **int** \rightarrow **real**. Such distributions on higher-order objects are required to model random processes, or Bayesian priors built from devices such as Dirichlet processes. This is probably the first time that examples of distributions on higher-order objects are given, backed by a dcpo-based semantics, and proved correct with respect to that semantics (Section 9.5).
- We give three operational semantics for our language in Section 10, Section 11 and 12, and we show that all of them are adequate with respect to the denotational semantics, under various, otherwise realistic, assumptions. *This study of adequacy has a nice consequence, which will end this list of contributions. One would expect a term M : **Dreal** to denote a distribution over real numbers, but our denotational semantics gives it a value as a continuous valuation over *intervals* of real numbers in \mathbb{R}_{\perp} , and that seems to do something different. We show that, under some benign assumptions, adequacy entails that the denotation of such a term M is really*

a distribution on ordinary real numbers, plus some ‘probability’ (not necessarily below 1) of non-termination (Corollary 11.6), as one would hope for.

Outline. We give some preliminary definitions in Section 2, where we also describe some of the challenges in more detail. In Section 3, we recapitulate the construction and basic properties of the monad \mathbf{V} of continuous valuations. The commutativity of that monad is equivalent to a form of the Fubini-Tonelli theorem, which in its most basic form expresses an interchange property for double integrals. Rather subtly, and perhaps paradoxically, such a Fubini-Tonelli theorem is easy to obtain for the \mathbf{V} monad on the category \mathbf{Top} of topological spaces, but is an open problem on the subcategory \mathbf{Dcpo} of dcpos. We will explain the issue in Section 4, where we will see that everything boils down to the fact that products in \mathbf{Dcpo} are in general different from products in the larger category \mathbf{Top} .

In order to obtain a Fubini-Tonelli theorem on \mathbf{Dcpo} , we may opt to restrict to, say, continuous dcpos, but this runs into some trouble, as we will have seen in Section 2. Our solution is much simpler: we restrict continuous valuations to a submonad \mathbf{V}_m of what we call *minimal* valuations, and we show that \mathbf{V}_m is a commutative monad on \mathbf{Dcpo} in Section 5. Now, since we will restrict our valuations to be minimal, doesn’t this exclude some interesting continuous distributions? And indeed that will seem to be the case: as we have already mentioned, we will show that Lebesgue valuation, seen as a continuous valuation on \mathbb{R} , is not minimal. But (and again, the shift is subtle), it *is* minimal on the domain \mathbb{R}_\perp that serves to do exact real arithmetic. In fact, as we show in Section 6, every measure on \mathbb{R} gives rise to a minimal valuation on \mathbb{R}_\perp . We will even give a simple, explicit description of the corresponding minimal valuation as a supremum of a countable chain of simple valuations.

Having done all this preliminary work, we introduce a higher-order statistical programming language which we call ISPCF for *Interval Statistical PCF*, with continuous distributions and soft constraints, as well as full recursion, in Section 7. Its denotational semantics is somehow straightforward, considering our preparatory steps, and is given in Section 8. In order to get a grasp of what we can express in ISPCF, and more importantly, how one can reason about ISPCF programs using that semantics, we provide a rather extensive list of examples in Section 9. This culminates with non-trivial examples of distributions on some higher-order data types.

Finally, we will explore a few operational semantics for ISPCF, paving the way for formally verified implementations. (We will not address implementations per se in this paper.) The first operational semantics for ISPCF we give is similar to some other earlier proposals [10, 17, 81], except that our real numbers are exact reals, not true reals, and that our transition function is continuous, not just measurable; it is given in Section 10. It is not too hard to give another operational semantics which works with *true* reals, and which is therefore even closer to the operational semantics given in [10, 17, 81]; this is the *precise* operational semantics of Section 11. Finally, we give a sampling-based operational semantics in Section 12. The name “sampling-based” is by analogy with early work by [62], and with one of the semantics of [10]. That semantics is meant to be one step closer to an implementation: instead of drawing real numbers at random with respect to arbitrary measures on \mathbb{R} , the sampling-based semantics draws bits independently at random, and derives certain continuous distributions from those bits. In each case, we show that our operational semantics are sound and adequate with respect to our denotational semantics.

We finish by reviewing related work in Section 13, and by concluding in Section 14.

2 PRELIMINARIES, CHALLENGES

We refer to [6, 67] for basics of measure theory, and to [1, 27, 31] for basics of domain theory and topology.

2.1 Measure theory

A σ -algebra on a set X is a collection of subsets closed under countable unions and complements. A *measurable space* X is a set with a σ -algebra Σ_X . The elements of Σ_X are usually called the *measurable subsets* of X .

A *measure* μ on X is a σ -additive map from Σ_X to $\overline{\mathbb{R}}_+$, where $\overline{\mathbb{R}}_+$ is the set of extended non-negative real numbers $\mathbb{R}_+ \cup \{+\infty\}$. We will agree that $0 \cdot (+\infty) = 0$. (This makes multiplication on $\overline{\mathbb{R}}_+$ Scott-continuous, see below.) The property of σ -additivity means that, for every countable family of pairwise disjoint sets E_n , $\mu(\bigcup_n E_n) = \sum_n \mu(E_n)$. (Here n ranges over any subset of \mathbb{N} , possibly empty.)

A *measurable map* $f: X \rightarrow Y$ between measurable spaces is a map such that $f^{-1}(E) \in \Sigma_X$ for every $E \in \Sigma_Y$. The *image measure* $f[\mu]$ of a measure μ on X is defined by $f[\mu](E) \stackrel{\text{def}}{=} \mu(f^{-1}(E))$.

The σ -algebra $\Sigma(A)$ *generated by* a family A of subsets of X is the the smallest σ -algebra containing A . The *Borel σ -algebra* on a topological space is the σ -algebra generated by its topology. The *standard topology* on $\overline{\mathbb{R}}_+$ is generated by the intervals $[0, b[$, $]a, b[$ and $]a, +\infty[$, with $0 < a < b < +\infty$. Its Borel σ -algebra is also generated by just the intervals $]a, +\infty[$ (the Scott-open subsets, see below). Hence a measurable map $h: X \rightarrow \overline{\mathbb{R}}_+$ is a map such that $h^{-1}(]t, +\infty[) \in \Sigma_X$ for every $t \in \mathbb{R}_+$. Its *Lebesgue integral* can be defined elegantly through *Choquet's formula*: $\int_X h \, d\mu \stackrel{\text{def}}{=} \int_0^{+\infty} \mu(h^{-1}(]t, +\infty[)) \, dt$, where the right-hand integral is an ordinary Riemann integral.

This formula makes the following *change-of-variables formula* an easy observation: for every measurable map $f: X \rightarrow Y$, for every measurable map $h: Y \rightarrow \overline{\mathbb{R}}_+$, $\int_Y h \, df[\mu] = \int_X (h \circ f) \, d\mu$.

There is a unique measure λ on \mathbb{R} such that $\lambda(]a, b[) = b - a$ for every open bounded interval $]a, b[$. This measure is called *Lebesgue measure*.

2.2 Domain theory and topology

A *dcpo* is a poset in which every directed family D has a supremum $\sup D$. A prime example is $\overline{\mathbb{R}}_+$, with the usual ordering. Another example is \mathbb{IR} , the poset of closed intervals $[a, b]$ with $a, b \in \mathbb{R}$ and $a \leq b$, ordered by reverse inclusion \supseteq . Every directed family $([a_i, b_i])_{i \in I}$ in \mathbb{IR} has a supremum $\bigcap_{i \in I} [a_i, b_i] = [\sup_{i \in I} a_i, \inf_{i \in I} b_i]$. \mathbb{IR}_\perp is the *lift* of \mathbb{IR} , namely the dcpo obtained by adding a fresh element \perp below all others. In general, we define the lift X_\perp of a dcpo X similarly. In the case of \mathbb{IR}_\perp , we may equate \perp with the whole set \mathbb{IR} itself, so that \mathbb{IR}_\perp is still ordered by reverse inclusion. \mathbb{IR}_\perp will be the domain of interpretation of the type **real** of exact real numbers. Among them, we find the *total numbers* $a \in \mathbb{R}$, which we may equate with the maximal elements $[a, a]$ of \mathbb{IR} .

\mathbb{IR}_\perp is an example of a *pointed* dcpo, namely one that has a least element, which we will always write as \perp , and which we read as *bottom*.

The *standard topology* on \mathbb{R} is generated by the open intervals $]a, b[$, with $a < b$. The map $i: a \mapsto [a, a]$ is then a topological embedding of \mathbb{R} , with its standard topology, into \mathbb{IR} (or \mathbb{IR}_\perp) with its Scott topology (see below for the Scott topology). In other words, i is continuous, and every open subset U of \mathbb{R} is the inverse image of some Scott-open subset of \mathbb{IR} (resp., \mathbb{IR}_\perp) by i . Explicitly, $]a, b[$ is the inverse image of the Scott-open subset of intervals $[c, d]$ such that $a < c \leq d < b$. This allows us to consider \mathbb{R} as a subspace of \mathbb{IR} , resp. \mathbb{IR}_\perp .

We will also write \leq for the ordering on any poset. In the example of \mathbb{IR} or \mathbb{IR}_\perp , \leq is \supseteq . The *upward closure* $\uparrow A$ of a subset A of a poset X is $\{y \in X \mid \exists x \in A, x \leq y\}$. The *downward closure* $\downarrow A$ is defined similarly. A set A is *upwards closed* if and only if $A = \uparrow A$, and *downwards closed* if and only if $A = \downarrow A$. A subset U of a poset X is *Scott-open* if and only if it is upwards closed and, for every directed family D such that $\sup D$ exists and is in U , some element of D is in U already. The Scott-open subsets of a poset X form its *Scott topology*.

The *way-below* relation \ll on a poset X is defined by $x \ll y$ if and only if, for every directed family D with a supremum z , if $y \leq z$, then x is less than or equal to some element of D already. We write $\uparrow x$ for $\{y \in X \mid x \ll y\}$, and $\downarrow y$ for $\{x \in X \mid x \ll y\}$. A poset X is *continuous* if and only if $\downarrow x$ is directed and has x as supremum for every $x \in X$. A *basis* B of a poset X is a subset of X such that $\downarrow x \cap B$ is directed and has x as supremum for every $x \in X$. A poset X is continuous if and only if it has a basis (namely, X itself). A poset is ω -*continuous* if and only if it has a countable basis. Examples include $\overline{\mathbb{R}}_+$, with any countable dense subset (with respect to its standard topology), such as the rational numbers, or such as the *dyadic numbers* $k/2^n$ ($k, n \in \mathbb{N}$); or \mathbb{R} and \mathbb{R}_\perp , with the basis of intervals $[a, b]$ where a and b are both dyadic.

We write OX for the lattice of open subsets of a topological space X . This applies to dcpos X as well, which will always be considered with their Scott topology. The continuous maps $f: X \rightarrow Y$ between two dcpos coincide with the *Scott-continuous* maps, namely the monotonic (order-preserving) maps that preserve all directed suprema. We write $\mathcal{L}X$ for the space of continuous maps from a topological space X to $\overline{\mathbb{R}}_+$, the latter with its Scott topology, as usual. Such maps are usually called *lower semicontinuous*, or *lsc*, in the mathematical literature. Note that $\mathcal{L}X$, with the pointwise ordering, is a dcpo.

2.3 Continuous valuations and measures

There are several ways in which one can model probabilistic choice. The most classical one is through measures. A popular alternative used in domain theory is given by *continuous valuations* [44, 45]. A continuous valuation is a Scott-continuous map $\nu: OX \rightarrow \overline{\mathbb{R}}_+$ such that $\nu(\emptyset) = 0$ (*strictness*) and, for all $U, V \in OX$, $\nu(U \cup V) + \nu(U \cap V) = \nu(U) + \nu(V)$ (*modularity*). Canonical examples of continuous valuations on X are *Dirac valuations* δ_x for $x \in X$, where for each open subset U of X , $\delta_x(U) = 1$ if $x \in U$ and $\delta_x(U) = 0$, otherwise. A *continuous valuation ν is bounded* if $\nu(X) < +\infty$. A *probability (resp., subprobability) valuation is a continuous valuation ν on X such that $\nu(X) = 1$ (resp., ≤ 1)*. The set of all continuous valuations on X is denoted by $\mathbf{V}X$. We order $\mathbf{V}X$ by the *stochastic order* defined as $\nu_1 \leq \nu_2$ if and only if $\nu_1(U) \leq \nu_2(U)$ for all opens of X . The set $\mathbf{V}X$ is a dcpo in the stochastic order.

There is a notion of integral $\int_{x \in X} h(x) d\nu$, or briefly $\int h d\nu$, for every $h \in \mathcal{L}X$, which can again be defined by a Choquet formula. Tix [79, Satz 4.4] showed that the integral is a Scott-continuous bilinear form, namely:

- for every $\nu \in \mathbf{V}X$, the map $h \in \mathcal{L}X \mapsto \int h d\nu$ is Scott-continuous and linear, in the sense that $\int \alpha h d\nu = \alpha \int h d\nu$ for every $\alpha \in \mathbb{R}$ and $\int (h + h') d\nu = \int h d\nu + \int h' d\nu$, for all $h, h' \in \mathcal{L}X$;
- for every $h \in \mathcal{L}X$, the map $\nu \in \mathbf{V}X \mapsto \int h d\nu$ is Scott-continuous and linear, in a similar sense.

More generally, a *linear* map $G: \mathcal{L}X \rightarrow \overline{\mathbb{R}}_+$ satisfies $G(h + h') = G(h) + G(h')$ and $G(\alpha \cdot h) = \alpha \cdot G(h)$ for all $\alpha \in \mathbb{R}_+$, $h, h' \in \mathcal{L}X$. Conversely, any Scott-continuous linear map $G: \mathcal{L}X \rightarrow \overline{\mathbb{R}}_+$ is of the form $h \mapsto \int h d\nu$ for a unique continuous valuation ν , given by $\nu(U) \stackrel{\text{def}}{=} G(\chi_U)$, where χ_U is the characteristic map of U ($\chi_U(x) \stackrel{\text{def}}{=} 1$ if $x \in U$, 0 otherwise).

Continuous valuations and measures are pretty much the same thing on ω -continuous dcpos, namely on continuous dcpos with a countable basis [5]. This holds more generally on de Brecht's quasi-Polish spaces [11], a class of spaces that contains not only the ω -continuous dcpos from domain theory but also the Polish spaces from topological measure theory.

One can see this as follows. In one direction, Adamski's theorem states that every measure μ on a hereditarily Lindelöf space X is τ -smooth [3, Theorem 3.1], meaning that its restriction to the lattice of open subsets of X is a continuous valuation. A hereditarily Lindelöf space is a space whose subspaces are all Lindelöf, or equivalently a space

in which every family of open sets contains a countable subfamily with the same union. Every second-countable space is hereditarily Lindelöf, and that includes all quasi-Polish spaces. In the other direction, every continuous valuation on an LCS-complete space extends to a Borel measure [12, Theorem 1.1]. An *LCS-complete* space is a space that is homeomorphic to a G_δ subset of a locally compact sober space. Every quasi-Polish space is LCS-complete; in fact, the quasi-Polish spaces are exactly the second-countable LCS-complete spaces [12, Theorem 9.5].

The relation between continuous valuations and measures is not one-to-one. Every τ -smooth measure on X restricts to exactly one continuous valuation on the lattice of open subsets of X , but a continuous valuation may extend to several measures on the Borel σ -algebra of X : the counting measure on \mathbb{R} , which maps every finite measurable subset to its cardinality and every infinite measurable subset to $+\infty$, and the infinite measure, which maps every non-empty measurable subset to $+\infty$, both restrict to the same continuous valuation on the open subsets of \mathbb{R} .

However, measure extensions of continuous valuations ν on X are unique in case X is locally compact and sober and ν is locally finite [4, Theorem 4.12]—the latter meaning that every point has an open neighborhood U such that $\nu(U) < +\infty$. Hence, on second-countable locally compact sober spaces, locally finite continuous valuations and locally finite measures are in one-to-one-correspondence.

2.4 Is there any trouble with the probabilistic powerdomain?

The probabilistic powerdomain, namely the $\text{dcpo } \mathbf{V}X$ of all continuous valuations on a space X , ordered pointwise, is known to have its problems [46]. Precisely, there is no known Cartesian-closed category of continuous dcpo s that is closed under the \mathbf{V} functor. It is sometimes believed that this means that domain theory cannot be used to give semantics to higher-order probabilistic languages. This would be wrong: the category \mathbf{Dcpo} of *all* dcpo s, not just the continuous dcpo s, is Cartesian-closed and closed under the \mathbf{V} functor [44, 45].

Continuity is not required to prove, say, soundness and adequacy theorems using logical relations, as one realizes by reading the relevant parts of [77], and as we will do in Section 10.2. But it is required to obtain a form of the Fubini-Tonelli theorem, or, in categorical terms, to turn \mathbf{V} into a *commutative* monad. (See [52, Theorem 9.2] for the relation between the two notions.) Commutativity is important in applications, as we will briefly discuss in Remark 8.1, and is a key ingredient of the semantics of [81]. We *do* obtain a commutative monad, without any need for continuity, by a simple trick based on inductive closures (Definition 5.1, Section 5).

2.5 Scoring, and density functions

Compared to ordinary probabilistic languages, statistical programming languages aim to offer the possibility of computing conditional distributions. This runs into questions on non-computability [2]. At this point, we note that there are well-established theories of computable probability distributions, notably on computable metric spaces [25], on more general computable topological spaces [65], and also based on Weihrauch’s type two theory of effectivity, also known as *TTE* [83], see [68, 84]. In the absence of probabilities, Schulz [69] shows that the functions from $[0, 1]^n$ to $[0, 1]$ that are computable in the sense of TTE and of RealPCF coincide.

In practice, a number of algorithms are implemented to compute certain special conditional distributions, with a fallback strategy based on one form or another of rejection sampling, or with a *scoring* mechanism, which allows one to give more or less importance to specific outcomes.

This scoring mechanism is typically implemented through a primitive called **score** [10, 17, 76, 81]. Roughly, the effect of **score** M , where $M: \mathbf{real}$ evaluates to a non-negative real number α , is to multiply the ‘probability’ of the current computation branch by α —making it a measure rather than a probability, whence the quotes.

As a consequence, **score** can be used to build new measures $g \cdot \mu$ from a measure μ and a density function g . The measure $g \cdot \mu$ is defined by $(g \cdot \mu)(E) \stackrel{\text{def}}{=} \int_X \chi_E(x)g(x) d\mu$, and is sometimes written as $g d\mu$. Indeed, in case x is drawn at random with respect to some measure μ , writing **score** $g(x); M(x)$ will have the effect of executing $M(x)$ as though x had been drawn with probability multiplied by $g(x)$, namely as though it had been drawn at random with respect to $g \cdot \mu$. This interpretation conforms to intuition if g really is a density function, namely if $\int_X g(x) d\mu = 1$, in which case $g \cdot \mu$ is a probability distribution. In general, however, $g \cdot \mu$ will be a measure. As a simple, but extreme example, **score** 0 annihilates the effect of the current computation. If x is drawn with a measure whose total mass is, say, $\pi/4$, then **score**($4/\pi$) will renormalize the measure to a probability distribution. Other uses of **score** include soft conditioning and Bayesian fitting, as illustrated in [81].

3 MONADS OF CONTINUOUS VALUATIONS

We will describe probabilistic effects by following Moggi's seminal work on monads [59, 60]. We use Manes' presentation of monads [56]: a monad (T, η, \dashv) on a category \mathbf{C} is a function T mapping objects of \mathbf{C} to objects of \mathbf{C} , a collection of morphisms $\eta_X: X \rightarrow TX$, one for each object X of \mathbf{C} , and called the *unit*, and for every morphism $f: X \rightarrow TY$, a morphism $f^\dagger: TX \rightarrow TY$ called the *extension* of f ; those are required to satisfy the axioms:

- (1) $f^\dagger \circ \eta_X = f$;
- (2) $\eta_X^\dagger = \text{id}_{TX}$;
- (3) $(g^\dagger \circ f)^\dagger = g^\dagger \circ f^\dagger$.

Then T extends to an endofunctor, acting on morphisms through $Tf = (\eta_Y \circ f)^\dagger$. Proposition 3.2 below is due to Jones [44, Theorem 4.5]. Her definition of the integral was different, and she implicitly restricted valuations to subprobability valuations. A similar statement is due to Kirch [50, Satz 6.1], for continuous dcpos instead of general dcpos. Tix [79] was probably the first to use the Choquet formula in this context. Her study was also restricted to continuous dcpos.

We define $\eta_X: X \rightarrow \mathbf{V}X$ as mapping $x \in X$ to δ_x . For every Scott-continuous map $f: X \rightarrow \mathbf{V}Y$, for every $\mu \in \mathbf{V}X$, for every Scott-open subset W of Y , we define:

$$f^\dagger(\mu)(W) \stackrel{\text{def}}{=} \int_{x \in X} f(x)(W) d\mu. \quad (1)$$

The following lemma and proposition are proved exactly as in most of the references we have just cited, hence their proofs are omitted.

LEMMA 3.1. *For all dcpos X and Y , for every Scott-continuous map $f: X \rightarrow \mathbf{V}Y$,*

- (i) *the map η_X is Scott-continuous;*
- (ii) *the map f^\dagger is Scott-continuous from $\mathbf{V}X$ to $\mathbf{V}Y$;*
- (iii) *for every $\mu \in \mathbf{V}X$, for every $g \in \mathcal{L}Y$,*

$$\int_{y \in Y} g(y) df^\dagger(\mu) = \int_{x \in X} \left(\int_{y \in Y} g(y) df(x) \right) d\mu. \quad (2)$$

- (iv) *For every $h \in \mathcal{L}X$, $\int_{x' \in X} h(x') d\delta_x = h(x)$.*

PROPOSITION 3.2. *The triple $(\mathbf{V}, \eta, \dashv)$ is a monad on the category \mathbf{Dcpo} of dcpos and Scott-continuous maps. For every Scott-continuous map $f: X \rightarrow Y$, for every $\mu \in \mathbf{V}X$, $\mathbf{V}f(\mu)$ is the image valuation $f[\mu]$, defined by $f[\mu](V) \stackrel{\text{def}}{=} \mu(f^{-1}(V))$, for every $V \in \mathcal{O}Y$.*

4 THE FUBINI-TONELLI THEOREM, AND ITS UNSOLVED SUBTLE VARIANT ON \mathbf{Depo}

Jones proved a form of the Fubini-Tonelli theorem for bounded continuous valuations on arbitrary topological spaces [44, Theorem 3.17]. The result extends to general, possibly unbounded continuous valuations, but one needs a new proof, since Jones' arguments do not extend.

As we will see in more detail later, this proposition turns \mathbf{V} into a commutative monad on \mathbf{Top} , and that is a basic requirement for being able to say that drawing two objects at random independently can be done in any order.

PROPOSITION 4.1 (FUBINI-TONELLI FOR CONTINUOUS VALUATIONS ON \mathbf{Top}). *Let X and Y be two topological spaces, $\mu \in \mathbf{V}X$, $\nu \in \mathbf{V}Y$. There is a unique product valuation $\mu \times \nu$ on $X \times Y$ such that, for every $U \in \mathcal{O}X$ and for every $V \in \mathcal{O}Y$, $(\mu \times \nu)(U \times V) = \mu(U) \cdot \nu(V)$. For every $f \in \mathcal{L}(X \times Y)$,*

$$\begin{aligned} \int_{(x,y) \in X \times Y} f(x,y) d(\mu \times \nu) &= \int_{x \in X} \left(\int_{y \in Y} f(x,y) d\nu \right) d\mu \\ &= \int_{y \in Y} \left(\int_{x \in X} f(x,y) d\mu \right) d\nu. \end{aligned}$$

PROOF. (Sketch.) That the maps $x \in X \mapsto \int_{y \in Y} f(x,y) d\nu$ and $y \in Y \mapsto \int_{x \in X} f(x,y) d\mu$ are lower semicontinuous, hence that the double integrals make sense, can be proved as in Jones' proof [44, Theorem 3.17]. We proceed with uniqueness. We assume any continuous valuation ξ on $X \times Y$ such that $\xi(U \times V) = \mu(U) \cdot \nu(V)$ for all $U \in \mathcal{O}X$ and $V \in \mathcal{O}Y$. We call any such product $U \times V$ an *open rectangle*. The value of ξ on finite unions $W \stackrel{\text{def}}{=} \bigcup_{i \in J} U_i \times V_i$ of open rectangles is determined uniquely: either $\mu(U_i) \cdot \nu(V_i) = +\infty$ for some $i \in J$, and then $\xi(W)$ must be equal to $+\infty$, by monotonicity; or $\xi(W)$ must be equal to $\sum_{K \neq \emptyset, K \subseteq J} (-1)^{|K|+1} \xi(\bigcap_{i \in K} U_i \times V_i)$ by the inclusion-exclusion formula (an easy consequence of modularity), showing that $\xi(W)$ is again determined uniquely. Finally, by definition of the product topology, every open subset W of $X \times Y$ is a union $\bigcup_{i \in I} U_i \times V_i$ of open rectangles, hence a directed union of finite unions $\bigcup_{i \in J} U_i \times V_i$ (where J ranges over the finite subsets of I); $\xi(W)$ is then determined uniquely, since ξ is Scott-continuous.

For the existence part, the easiest route is to consider the two maps $G, G': \mathcal{L}(X \times Y) \rightarrow \overline{\mathbb{R}}_+$ defined by:

$$\begin{aligned} G(f) &\stackrel{\text{def}}{=} \int_{x \in X} \left(\int_{y \in Y} f(x,y) d\nu \right) d\mu \\ G'(f) &\stackrel{\text{def}}{=} \int_{y \in Y} \left(\int_{x \in X} f(x,y) d\mu \right) d\nu. \end{aligned}$$

We check that G and G' are Scott-continuous, linear maps, and are therefore integral functionals for unique continuous valuations ξ and ξ' on $X \times Y$, respectively. For every open rectangle $U \times V$, $\xi(U \times V) = G(\chi_{U \times V})$ is equal to $\mu(U) \cdot \nu(V)$, and similarly for $\xi'(U \times V)$. By the uniqueness part, $\xi = \xi'$, so $G = G'$. We write $\mu \times \nu$ for ξ , and the theorem is proved. \square

Remark 4.1. It is well-known that the Fubini-Tonelli type equation does not hold on the category \mathbf{Meas} of measurable sets and measurable maps; typical counter-examples are built by comparing the nested integrals of the characteristic function χ_Δ of the diagonal subset $\Delta = \{(x,x) \mid x \in [0,1]\}$ of $[0,1] \times [0,1]$, against the Lebesgue measure and the counting measure on $[0,1]$. The reason why Proposition 4.1 holds is that we only consider integrands that are lower-semicontinuous on $X \times Y$. Indeed, the function χ_Δ is measurable, but far from being lower-semicontinuous.

Remark 4.2. This proof is the core of several similar proofs. One of the closest is due to Vickers [82], who proves a similar theorem on the category of locales instead of \mathbf{Top} . Although localic theorems usually generalize purely

topological theorems, this one does not, because locale products do not coincide with products in **Top** in general [41, Theorem 2].

Since every dcpo can be seen as a topological space with its Scott topology, it would seem that we would obtain a Fubini-Tonelli theorem for Scott-continuous maps and continuous valuations on arbitrary dcpos, as a special case of Proposition 4.1.

This is not the case, but the reason is subtle. As with locales, products in **Dcpo** do not usually coincide with products in **Top**. Explicitly, let us write X_σ for the topological space obtained by equipping a dcpo X with its Scott topology. Then the topology on $(X \times Y)_\sigma$ (where \times is dcpo product) is finer, and in general strictly finer, than the product topology on $X_\sigma \times Y_\sigma$. (See Exercise 5.2.16 of [31] for an example where it is strictly finer.) As a consequence, there are more, and generally, strictly more Scott-continuous maps from $X \times Y$ to $\overline{\mathbb{R}}_+$ than continuous maps from the topological product $X_\sigma \times Y_\sigma$ to $\overline{\mathbb{R}}_+$. The Fubini-Tonelli formula holds for functions of the second, smaller class, but it is unknown whether it holds for the first kind of functions. One exception is when either X or Y is *core-compact* (the Scott-opens form a continuous lattice), in which case the Scott and product topologies coincide on $X \times Y$ [27, Theorem II-4.13]; then we retrieve Jones' version of the Fubini-Tonelli theorem.

One may blame **Dcpo** for this state of affairs. Another possibility is to consider that we are considering too general a notion of continuous valuation. We will explore this avenue in the next section, by restricting to [what we call](#) minimal valuations.

5 MINIMAL VALUATIONS, AND FUBINI-TONELLI AGAIN

In this section we develop a commutative valuations monad on the category of dcpos and Scott-continuous maps. As mentioned in the introduction, this monad [was](#) also discovered by the authors of [42]. [Hence we will only give proof sketches.](#)

Definition 5.1 (Minimal valuations). A *simple valuation* is any finite linear combination $\sum_{i=1}^n r_i \delta_{x_i}$ of Dirac masses, with coefficients r_i in \mathbb{R}_+ . Let $\mathbf{V}_f X$ be the poset of simple valuations on the topological space X , and $\mathbf{V}_m X$ be the inductive closure of $\mathbf{V}_f X$ in $\mathbf{V}X$. The elements of $\mathbf{V}_m X$ are called the *minimal valuations* on X .

The *inductive closure* of a subset A of a dcpo Z is the smallest subset of Z that contains A and is closed under directed suprema. It is obtained by taking all directed suprema of elements of A , all directed suprema of elements obtained in this fashion, and proceeding this way transfinitely.

Remark 5.1. On a continuous dcpo X , every continuous valuation is a directed supremum of simple valuations. In fact, $\mathbf{V}X$ is a continuous dcpo with a basis of simple valuations, as showed by Jones [44], at least in the case of subprobability valuations. (The general theorem can be found as Theorem IV-9.16 of [27].) Hence, in particular, continuous valuations and minimal valuations agree on every continuous dcpo. One may wonder whether all continuous valuations on a dcpo are minimal. We give a counterexample to this claim in [35].

5.1 The monad \mathbf{V}_m of minimal valuations

We will now show that \mathbf{V}_m defines a submonad of \mathbf{V} . To this end, we need to know more about inductive closures. A *d-closed subset* of a dcpo Z is a subset C such that the supremum of every directed family of elements of C , taken in Z , is in C . The d-closed subsets form the closed subsets of a topology called the *d-topology* [49, Section 5], and the inductive closure of a subset A coincides with its *d-closure* $cl_d(A)$, namely its closure in the d-topology.

We note that every Scott-continuous map is continuous with respect to the underlying d-topologies. This is easily checked, or see [49, Lemma 5.3]. In particular, image of d-closures are included in d-closures of images: (*) for every Scott-continuous map $f: \mathbf{V}X \rightarrow \mathbf{V}Y$, for every $A \subseteq \mathbf{V}X$, $f(\text{cl}_d(A)) \subseteq \text{cl}_d(f(A))$. It is then easy to show that, for every space X , $\mathbf{V}_m X$ is closed under addition and multiplication by elements of $\overline{\mathbb{R}}_+$, as computed in the larger space $\mathbf{V}X$. For example, let us look at addition $+$. Fixing a simple valuation μ , the map $f_\mu: v \in \mathbf{V}X \mapsto \mu + v$ is Scott-continuous, and maps simple valuations to simple valuations. By (*), f_μ maps all elements of $\text{cl}_d(A) = \mathbf{V}_m X$ to $\text{cl}_d(f_\mu(A)) \subseteq \text{cl}_d(\mathbf{V}_f X) = \mathbf{V}_m X$. We reason similarly by fixing a minimal valuation ν in order to show that the map $\mu \in \mathbf{V}X \mapsto \mu + \nu = f_\mu(\nu)$ maps simple valuations to minimal valuations, and therefore minimal valuations to minimal valuations.

Given any Scott-continuous map $f: X \rightarrow \mathbf{V}_m Y$, where X is a dcpo and Y is a topological space, it follows that $f^\dagger(\sum_{i=1}^n r_i \delta_{x_i}) = \sum_{i=1}^n r_i f(x_i)$ is a minimal valuation. Hence f^\dagger maps simple valuations to minimal valuations, and therefore also minimal valuations to minimal valuations, by (*).

We observe that $\eta_X(x) = \delta_x$ is in $\mathbf{V}_f X \subseteq \mathbf{V}_m X$ for every dcpo X , and every $x \in X$. We then use Proposition 3.2 and we obtain the following.

PROPOSITION 5.2. *The triple $(\mathbf{V}_m, \eta, _^\dagger)$ is a monad on the category of dcpos and Scott-continuous maps.*

5.2 Tensorial strengths

A tensorial strength for a monad $(T, \eta, _^\dagger)$ is a collection t of morphisms $t_{X,Y}: X \times TY \rightarrow T(X \times Y)$, natural in X and Y , satisfying certain coherence conditions (which we omit, see [60].) We then say that $(T, \eta, _^\dagger, t)$ is a *strong* monad. We will satisfy ourselves with the following result. By [60, Proposition 3.4], in a category with finite products and enough points, if one can find morphisms $t_{X,Y}: X \times TY \rightarrow T(X \times Y)$ for all objects X and Y such that $t_{X,Y} \circ \langle x, \nu \rangle = T(\langle x \circ!, \text{id}_Y \rangle) \circ \nu$, then the collection of those morphisms is the unique tensorial strength. A category with a terminal object 1 has *enough points* if and only if, for any two morphisms $f, g: X \rightarrow Y$, $f = g$ if and only if for every $x: 1 \rightarrow X$, $f \circ x = g \circ x$.

The category \mathbf{Dcpo} of dcpos has finite products, and has enough points. Specializing the above to $T \stackrel{\text{def}}{=} \mathbf{V}$, the formula for $t_{X,Y}$ reads: for every $x \in X$, for every $\nu \in \mathbf{V}Y$, for every $W \in \mathcal{O}(X \times Y)$, $t_{X,Y}(x, \nu)(W) = \nu(\{y \in Y \mid (x, y) \in W\})$. Rewriting this as $t_{X,Y}(x, \nu)(W) = \int_{y \in Y} \chi_W(x, y) d\nu$, hence $t_{X,Y}(x, \nu) = \mathbf{V}(\lambda y \in Y. (x, y))(v)$, we retrieve formulae already given by Jones [44, Section 4.3], and which show immediately that the map $t_{X,Y}$ is well-defined and Scott-continuous.

If ν is a simple valuation $\sum_{i=1}^n r_i \delta_{x_i}$, then $t_{X,Y}(x, \nu) = \sum_{i=1}^n r_i \delta_{(x, x_i)}$, so $t_{X,Y}(x, _)$ maps simple valuations to simple valuations, and using (*) as above, minimal valuations to minimal valuations. It follows:

PROPOSITION 5.3. *$(\mathbf{V}, \eta, _^\dagger, t)$ and $(\mathbf{V}_m, \eta, _^\dagger, t)$ are strong monads on \mathbf{Dcpo} .*

5.3 Minimal valuations form a commutative monad on \mathbf{Dcpo} , or Fubini-Tonelli again

We now show that \mathbf{V}_m is a *commutative* monad on \mathbf{Dcpo} . The corresponding result is unknown for \mathbf{V} . Equivalently, the Fubini-Tonelli theorem holds for minimal valuations. In order to prove it, we use the following simple lemma.

LEMMA 5.4. *Two morphisms $f, g: X \rightarrow Y$ in \mathbf{Dcpo} that coincide on $A \subseteq X$ also coincide on $\text{cl}_d(A)$.*

PROOF. Let $B \stackrel{\text{def}}{=} \{x \in X \mid f(x) = g(x)\}$. Since f and g preserve directed suprema, B is d-closed. By assumption, A is included in B , so B also contains $\text{cl}_d(A)$. \square

THEOREM 5.5 (FUBINI-TONELLI FOR MINIMAL VALUATIONS ON \mathbf{Dcpo}). *Let X, Y be two dcpos, $f \in \mathcal{L}(X \times Y)$, $\mu \in \mathbf{V}X$, and $\nu \in \mathbf{V}Y$. If μ or ν is minimal, then:*

$$\int_{x \in X} \left(\int_{y \in Y} f(x, y) d\nu \right) d\mu = \int_{y \in Y} \left(\int_{x \in X} f(x, y) d\mu \right) d\nu. \quad (3)$$

PROOF. It is clear that the two sides of the equation coincide when μ (or ν) is a simple valuation. The result then follows from Lemma 5.4. \square

Given a tensorial strength t for a monad T on a category with finite products, there is a dual tensorial strength t' , where $t'_{X,Y}: TX \times Y \rightarrow T(X \times Y)$, obtained by swapping the two arguments, applying $t_{X,Y}$, and then swapping back the roles of X and Y . Here $t'_{X,Y}(\mu, \nu)(W) = \int_{x \in X} \chi_W(x, y) d\mu$. We can then define two morphisms from $TX \times TY$ to $T(X \times Y)$, namely $t'_{X,Y} \circ t_{TX,Y}$ and $t'_{X,Y} \circ t'_{X,TY}$. The monad T is *commutative* when they coincide. The connection between commutative monads and Fubini-Tonelli-like theorems is made explicit by Kock [52].

Computing $t'_{X,Y} \circ t'_{X, \mathbf{V}Y}$ and $t'_{X,Y} \circ t_{\mathbf{V}X, Y}$ shows that the following is a mere restatement of Theorem 5.5.

PROPOSITION 5.6. *Let X, Y be two dcpos. The maps $t'_{X,Y} \circ t'_{X, \mathbf{V}Y}$ and $t'_{X,Y} \circ t_{\mathbf{V}X, Y}$ coincide on those pairs $(\mu, \nu) \in \mathbf{V}X \times \mathbf{V}Y$ such that $\mu \in \mathbf{V}_m X$ or $\nu \in \mathbf{V}_m Y$.*

COROLLARY 5.7. $(\mathbf{V}_m, \eta, _ \dashv, t)$ is a commutative monad on \mathbf{Dcpo} .

Remark 5.2. Strictly speaking, the Fubini-Tonelli theorem appears as being more general than Equation 3, since it also states the existence of a product measure. However, the latter is derivable. This is an instance of the general fact that commutative monads are equivalent to monoidal monads [51, Proposition 1.5, equivalence between (iii) and (v)]. Explicitly, in our case, we obtain a minimal product valuation as follows. We write \otimes for the morphism $t'_{X,Y} \circ t_{\mathbf{V}X, Y}$ ($= t'_{X,Y} \circ t'_{X, \mathbf{V}Y}$) from $\mathbf{V}_m X \times \mathbf{V}_m Y$ to $\mathbf{V}_m(X \times Y)$, as with any commutative monad. Then, for all $\mu \in \mathbf{V}_m X$ and $\nu \in \mathbf{V}_m Y$, $\otimes(\mu, \nu)$, which we prefer to write as $\mu \otimes \nu$, is in $\mathbf{V}_m(X \times Y)$. Looking back at the computations we have done during the proof of Proposition 5.6,

$$\begin{aligned} (\mu \otimes \nu)(W) &= \int_{x \in X} \left(\int_{y \in Y} \chi_W(x, y) d\nu \right) d\mu \\ &= \int_{y \in Y} \left(\int_{x \in X} \chi_W(x, y) d\mu \right) d\nu \end{aligned}$$

for every $W \in \mathcal{O}(X \times Y)$. An easy computation, based on (2), yields that, for every $f \in \mathcal{L}(X \times Y)$, $\int_{(x,y) \in X \times Y} f(x, y) d(\mu \otimes \nu)$ is equal to any of the double integrals of (3).

As an additional benefit of the categorical approach, we obtain that the map $\otimes: (\mu, \nu) \mapsto \mu \otimes \nu$ is Scott-continuous from $\mathbf{V}_m X \times \mathbf{V}_m Y$ to $\mathbf{V}_m(X \times Y)$.

Remark 5.3. A general categorical method for obtaining submonads is given by Kammar and McDermott in [48, Section 2.2] by using orthogonal factorisation systems, which eventually leads to the commutative monad T considered in [81]. There is a definite air de famille between our construction and theirs. However, we have not been able to use it to retrieve our commutative monad of minimal valuations on \mathbf{Dcpo} .

6 MEASURES ON \mathbb{R} , MINIMAL VALUATIONS ON \mathbb{IR}

All this is good, but aren't we restricting continuous valuations too much by only considering minimal valuations? And indeed, Lebesgue valuation is not minimal on \mathbb{R} , as we will soon see. But it is minimal on \mathbb{IR} and on \mathbb{IR}_\perp . In fact, we will see that every measure μ on \mathbb{R} is represented by a minimal valuation on \mathbb{IR} and on \mathbb{IR}_\perp , in the following sense.

To be more precise, since \mathbb{R} is second-countable hence hereditarily Lindelöf, every measure μ on \mathbb{R} restricts to a continuous valuation on \mathbb{R} . Its image valuation $i[\mu]$ by the embedding $i: x \mapsto [x, x]$ of \mathbb{R} into \mathbb{IR} (or \mathbb{IR}_\perp) is then a continuous valuation on \mathbb{IR} (resp., \mathbb{IR}_\perp). While the former is rarely minimal, we will see that the latter always is.

Let us call *valuation* on X any strict, modular, monotonic map from OX to $\overline{\mathbb{R}}_+$; namely, we forgo the continuity requirement. A valuation v on X is *point-continuous* if and only if for every open subset U of X , for every $r \in \mathbb{R}_+$ such that $r < v(U)$, there is a finite subset A of U such that, for every open neighborhood V of A , $v(V) > r$. The notion is due to Heckmann [37]. His main achievement was to show that the space $\mathbf{V}_p X$ of point-continuous valuations over X , with the weak topology (which we will not define), is a sobrification of the space of simple valuations, also with the weak topology. This can be used to show that there is also a commutative monad $(\mathbf{V}_p, \eta, \dashv, t)$ on \mathbf{Dcpo} , but we will not show this here. We will use the following results by Heckmann: every point-continuous valuation is continuous; every simple valuation is point-continuous; $\mathbf{V}_p X$ is sober, in particular it is a dcpo under the pointwise ordering; in particular, every minimal valuation is point-continuous.

Let λ be the Lebesgue measure on \mathbb{R} , or ambiguously, *the Lebesgue valuation, obtained by restriction to the open sets. This cannot cause confusion, since λ is locally finite.*

LEMMA 6.1. *The Lebesgue valuation λ on \mathbb{R} is not point-continuous, hence not minimal.*

PROOF. Let U be any non-empty open subset of \mathbb{R} . Then $\lambda(U) > 0$. Let us pick any $r \in \mathbb{R}_+$ such that $r < \lambda(U)$. For every finite subset A of U , say of cardinality n , the open set $V \stackrel{\text{def}}{=} \bigcup_{x \in A}]x - \epsilon, x + \epsilon[$, where $\epsilon > 0$ is chosen so that $2n\epsilon < r$, is an open neighborhood of A but $\lambda(V) \leq \sum_{x \in A} \lambda(]x - \epsilon, x + \epsilon[) = 2n\epsilon < r$. \square

PROPOSITION 6.2. *For every measure μ on \mathbb{R} , and writing again μ for the valuation it induces by restriction to $O\mathbb{R}$, the image valuation $i[\mu]$ is a minimal valuation on \mathbb{IR} (resp., \mathbb{IR}_\perp).*

PROOF. \mathbb{IR} (resp., \mathbb{IR}_\perp) is a continuous dcpo. Hence $\mathbf{V}(\mathbb{IR})$ (resp., $\mathbf{V}(\mathbb{IR}_\perp)$) is also a continuous dcpo, with a basis of simple valuations [27, Theorem IV-9.16]. In particular, $i[\mu]$ is a directed supremum of simple valuations, hence a minimal valuation. \square

Remark 6.1. The proof of Proposition 6.2 shows, more generally, that, given any hereditarily Lindelöf space X with a topological embedding i into some continuous dcpo P , for every Borel measure μ on X , $i[\mu]$ is a minimal valuation on P . Lawson showed that every Polish space X , not just \mathbb{R} , has this property [54].

\mathbb{IR} and \mathbb{IR}_\perp are even ω -continuous, namely, they have a countable basis. One can then show that $i[\mu]$ is the supremum of a countable chain of simple valuations (Exercise IV-9.29 of [27] helps, which says that on a ω -continuous domain, simple valuations with rational coefficients that also are supported on the countable basis form a basis for all continuous valuations). But we can also obtain such an explicit countable chain by elementary means. The following construction, which has some common points with the Riemann-Stieltjes integral, is uniform, and only requires the knowledge of $\mu(]a, b])$ for intervals $]a, b]$ with dyadic endpoints. A similar and more general framework of the construction, in the setting of locally compact Hausdorff topological spaces, can be found in [16].

Definition 6.3. A *partition* P (of \mathbb{R}) is a finite non-empty subset of \mathbb{R} . We write P as $\{a_1 < \dots < a_n\}$ in order to make both the elements and their order manifest, with $n \geq 1$. Every such partition induces a finite subset E_P of $\mathbb{I}\mathbb{R}$ defined by $E_P \stackrel{\text{def}}{=} \{[a_1, a_2], \dots, [a_{n-1}, a_n]\}$. If $n = 1$, then E_P is empty.

Let \mathcal{P} be the set of all partitions of \mathbb{R} . A partition Q *refines* P if and only if $P \subseteq Q$.

Given any measure μ on \mathbb{R} , and any partition $P \stackrel{\text{def}}{=} \{a_1 < \dots < a_n\}$ of \mathbb{R} , we let μ_P be the simple valuation $\sum_{i=2}^n \mu(\cdot)_{[a_{i-1}, a_i]}$ on $\mathbb{I}\mathbb{R}$ or on $\mathbb{I}\mathbb{R}_\perp$.

Remark 6.2. The coefficient of $\delta_{[a_{i-1}, a_i]}$ in μ_P is $\mu(\cdot)_{[a_{i-1}, a_i]}$, not $\mu([a_{i-1}, a_i])$. The point is that $\mu(\cdot)_{[a_{i-1}, a_i]} + \mu(\cdot)_{[a_i, a_{i+1}]} = \mu(\cdot)_{[a_{i-1}, a_{i+1}]}$. Choosing the coefficient to be $\mu([a_{i-1}, a_i])$ would work equally well.

Remark 6.3. If μ is a probability valuation on \mathbb{R} given by its cumulative distribution function F (namely, $F(t) \stackrel{\text{def}}{=} \mu(\cdot)_{(-\infty, t]}$), then μ_P is equal to $(F(a_2) - F(a_1)) \cdot \delta_{[a_1, a_2]} + \dots + (F(a_n) - F(a_{n-1})) \cdot \delta_{[a_{n-1}, a_n]}$, and represents the process of picking the interval $[a_{i-1}, a_i]$ with probability $F(a_i) - F(a_{i-1})$.

THEOREM 6.4. *Let μ be any measure on \mathbb{R} . Then:*

- (1) *for all partitions P and Q of \mathbb{R} , if $P \subseteq Q$ then $\mu_P \leq \mu_Q$;*
- (2) *$i[\mu]$ is the directed supremum of $(\mu_P)_{P \in \mathcal{P}}$, and is therefore a minimal valuation on $\mathbb{I}\mathbb{R}$;*
- (3) *for every dense subset D of \mathbb{R} , $i[\mu]$ is also the directed supremum of $(\mu_P)_{P \in \mathcal{P}_D}$, where \mathcal{P}_D is the collection of partitions $P \subseteq D$.*

PROOF. 1. Since Q refines P , Q is obtained by adding some points to P , and it suffices to show the claim when we add just one point. Let us write P as $\{a_1 < \dots < a_n\}$, and let $Q \stackrel{\text{def}}{=} P \cup \{a\}$, where $a \notin P$. If $a < a_1$, then μ_Q is equal to $\mu_P + \mu(\cdot)_{[a, a_1]}$; similarly if $a > a_n$. Otherwise, let j be the unique index such that $a_{j-1} < a < a_j$, $2 \leq j \leq n$. Then:

$$\begin{aligned} \mu_P &= \sum_{\substack{i=2 \\ i \neq j}}^n \mu(\cdot)_{[a_{i-1}, a_i]} \cdot \delta_{[a_{i-1}, a_i]} + \mu(\cdot)_{[a_{j-1}, a_j]} \cdot \delta_{[a_{j-1}, a_j]} \\ &= \sum_{\substack{i=2 \\ i \neq j}}^n \mu(\cdot)_{[a_{i-1}, a_i]} \cdot \delta_{[a_{i-1}, a_i]} + \mu(\cdot)_{[a_{j-1}, a]} \cdot \delta_{[a_{j-1}, a]} + \mu(\cdot)_{[a, a_j]} \cdot \delta_{[a, a_j]} \\ &\leq \sum_{\substack{i=2 \\ i \neq j}}^n \mu(\cdot)_{[a_{i-1}, a_i]} \cdot \delta_{[a_{i-1}, a_i]} + \mu(\cdot)_{[a_{j-1}, a]} \cdot \delta_{[a_{j-1}, a]} + \mu(\cdot)_{[a, a_j]} \cdot \delta_{[a, a_j]} = \mu_Q, \end{aligned}$$

where the latter inequality is justified by the fact that $[a_{j-1}, a]$ and $[a, a_j]$ are both larger than or equal to $[a_{j-1}, a_j]$, and that $x \mapsto \delta_x$ is monotonic.

Item 2 is a special case of item 3, with $D \stackrel{\text{def}}{=} \mathbb{R}$. Hence we prove item 3 directly. Let D be any dense subset of \mathbb{R} .

The family \mathcal{P}_D is directed under inclusion, since $\{d\} \in \mathcal{P}$ for any given $d \in D$, and since for all $P, Q \in \mathcal{P}_D$, $P \cup Q$ is in \mathcal{P}_D . By item 1, $(\mu_P)_{P \in \mathcal{P}_D}$ is therefore directed.

Let us fix an arbitrary Scott-open subset \mathcal{U} of $\mathbb{I}\mathbb{R}$ (resp., $\mathbb{I}\mathbb{R}_\perp$), and let $U \stackrel{\text{def}}{=} i^{-1}(\mathcal{U})$.

For every $P \stackrel{\text{def}}{=} \{a_1 < \dots < a_n\} \in \mathcal{P}_D$, $\mu_P(\mathcal{U}) \leq i[\mu](\mathcal{U})$. Indeed,

$$\begin{aligned} \mu_P(\mathcal{U}) &= \sum_{\substack{2 \leq i \leq n \\ [a_{i-1}, a_i] \in \mathcal{U}}} \mu([a_{i-1}, a_i]) \\ &= \mu \left(\bigcup_{\substack{2 \leq i \leq n \\ [a_{i-1}, a_i] \in \mathcal{U}}}]a_{i-1}, a_i[\right) && \text{since the sets }]a_{i-1}, a_i[\text{ are pairwise disjoint} \\ &\leq \mu(U) = i[\mu](\mathcal{U}). \end{aligned}$$

The last inequality is justified by the fact that for every i with $2 \leq i \leq n$ such that $[a_{i-1}, a_i] \in \mathcal{U}$, $[a_{i-1}, a_i]$ is included in U : for every $x \in [a_{i-1}, a_i]$, $[a_{i-1}, a_i]$ is below $[x, x]$ in \mathbb{R} (resp. \mathbb{R}_\perp), so $[x, x] \in \mathcal{U}$, meaning that $x \in U$.

In order to show that $\sup_{P \in \mathcal{P}_D} \mu_P(\mathcal{U}) \geq i[\mu](\mathcal{U})$, we consider any $r \in \mathbb{R}_+$ such that $r < i[\mu](\mathcal{U}) = \mu(U)$, and we will show that there is a partition $P \in \mathcal{P}_D$ such that $\mu_P(\mathcal{U}) > r$. For every $x \in U$, $[x, x]$ is in \mathcal{U} and is the supremum of the chain of elements $[x - \epsilon, x + \epsilon]$, $\epsilon > 0$. Therefore there is a positive real number ϵ_x such that $[x - \epsilon_x, x + \epsilon_x]$ is in \mathcal{U} . Since D is dense in \mathbb{R} , we can find a subinterval $[a_x, b_x]$ such that a_x and b_x are in D , and $a_x < x < b_x$. We note that $[a_x, b_x]$ is also in \mathcal{U} . In particular, $[a_x, b_x]$, hence also $]a_x, b_x[$, is included in U for every $x \in U$, so $U = \bigcup_{x \in U}]a_x, b_x[$. We write the latter as the directed union of the open sets $U_E \stackrel{\text{def}}{=} \bigcup_{x \in E}]a_x, b_x[$, where E ranges over the finite subsets of U . Since μ restricted to the open sets of \mathbb{R} is a continuous valuation, there is a finite subset E of U such that $r < \mu(U_E)$.

We let P be the collection of elements a_x and b_x , $x \in E$. This is an element of \mathcal{P}_D . Let us write P as $\{c_1 < \dots < c_n\}$. For every $x \in E$, $[a_x, b_x]$ is an element of \mathcal{U} , and is equal to $[c_{i_x}, c_{j_x}]$ for some indices i_x and j_x such that $1 \leq i_x < j_x \leq n$. Then $[c_{i-1}, c_i]$ is larger than or equal to $[a_x, b_x]$ for every i with $i_x + 1 \leq i \leq j_x$, hence is also in \mathcal{U} . It follows that $\mu_P(\mathcal{U}) = \sum_{\substack{2 \leq i \leq n \\ [c_{i-1}, c_i] \in \mathcal{U}}} \mu([c_{i-1}, c_i]) \geq \sum_{\substack{x \in E \\ i_x + 1 \leq i \leq j_x}} \mu([c_{i-1}, c_i])$. The latter is equal to $\mu(\bigcup_{\substack{x \in E \\ i_x + 1 \leq i \leq j_x}}]c_{i-1}, c_i[)$, since the intervals $]c_{i-1}, c_i[$ are pairwise disjoint. By definition of i_x and j_x , this is larger than or equal to $\mu(\bigcup_{x \in E}]a_x, b_x[)$, hence to $\mu(\bigcup_{x \in E}]a_x, b_x[) = \mu(U_E)$. Therefore $r < \mu_P(\mathcal{U})$, as desired. \square

COROLLARY 6.5. *Let μ be any measure on \mathbb{R} . For every $n \in \mathbb{N}$, let the partition P_n consist of all integer multiples of $1/2^n$ between $-n$ and n . The minimal valuation $i[\mu]$ is the supremum of the countable chain of simple valuations μ_{P_n} , $n \in \mathbb{N}$.*

PROOF. The family $(P_n)_{n \in \mathbb{N}}$ is a subfamily of \mathcal{P}_D , where D is the set of dyadic numbers. It is in fact cofinal: every element P of \mathcal{P}_D is refined by P_n for some $n \in \mathbb{N}$, namely for any n larger than every element of P , every opposite of an element of P , and such that every element of P is an integer multiple of $1/2^n$. It follows that $(\mu_P)_{P \in \mathcal{P}_D}$ and $(\mu_{P_n})_{n \in \mathbb{N}}$ have the same supremum, and that is $i[\mu]$ by Theorem 6.4, item 3. \square

We now see that each Borel measure μ on \mathbb{R} can be realized as a minimal valuation $i[\mu]$ on \mathbb{R}_\perp . Let us say something about the other direction. Every continuous valuation ν on \mathbb{R}_\perp is minimal and extends to a measure $\bar{\nu}$ on the Borel σ -algebra generated by Scott opens on \mathbb{R}_\perp [12]. The extension $\bar{\nu}$ is unique if ν is bounded in the sense that $\nu(\mathbb{R}_\perp) < \infty$. The measure $\bar{\nu}$ then restricts to a measure $\bar{\nu}|_A$ on any measurable subset A of \mathbb{R}_\perp . Hence on the subset $M = \{[x, x] \mid x \in \mathbb{R}\}$ of maximal elements, which is a G_δ subset of \mathbb{R}_\perp and homeomorphic to \mathbb{R} [16], $\bar{\nu}|_M$ is a measure. Moreover, if we start with a bounded Borel measure μ on \mathbb{R} , and realizes it as the minimal valuation $i[\mu]$ on \mathbb{R}_\perp , extends $i[\mu]$ to the measure $\bar{i}[\mu]$ on \mathbb{R}_\perp and then restricts to $\bar{i}[\mu]|_M$, one essentially gets μ back, up to the obvious homomorphism $x \mapsto [x, x]$ between \mathbb{R} and M . However, if we start with a bounded valuation ν on \mathbb{R}_\perp , and sequentially apply the extension, restriction, and push-forward the resulting valuation back to \mathbb{R}_\perp , one will not get back ν in general. Take δ_\perp on \mathbb{R}_\perp

for example. We will get the constantly zero valuation after applying the above operations, which is strictly smaller than δ_{\perp} .

We will see that our language ISPCF could compute minimal valuations of the form $i[\mu]$, where μ is any measure on \mathbb{R} . Clearly, these valuations could be equated with measures on reals. But one could also write programs (at type **real**) in ISPCF whose denotations are not of the form $i[\mu]$. A typical example is the program **ret** M , where M is any program of type **real** that diverges, according to our denotational semantics in Section 8, it denotes δ_{\perp} in $\mathbf{V}_m\mathbb{R}_{\perp}$. More on this matter is discussed at the end of Section 11, where we will see that, under reasonable assumptions on the constants we include in ISPCF, the only continuous valuations computed at type **real** are linear combinations of measures μ on \mathbb{R} (in the form of $i[\mu]$) and of non-termination, δ_{\perp} .

7 THE ISPCF CALCULUS

We now come to the description of our calculus. This is a variant of SFPC, a statistical variant of Fiore and Plotkin’s Fixed Point Calculus [23] due to Vákár, Kammar, and Staton [81]. It is also very close to the calculus PCFSS of [10] and to PPCF [17]. Just like the latter, but contrarily to the other languages we have mentioned, our calculus ISPCF (for *Interval* statistical PCF) is a call-by-name language.

Its algebra of types is as follows. One may naturally consider additional types, such as more complex, recursively defined data types.

$$\sigma, \tau, \dots ::= \mathbf{unit} \mid \mathbf{void} \mid \mathbf{int} \mid \mathbf{real} \mid \sigma + \tau \mid \sigma \times \tau \mid \sigma \rightarrow \tau \mid D\tau$$

The types of the form $D\tau$ are called *distribution types*. The function arrow \rightarrow associates to the right, so that $\sigma \rightarrow \tau \rightarrow \lambda$ abbreviates $\sigma \rightarrow (\tau \rightarrow \lambda)$. Compared to the algebra of types of SFPC, first, we keep a type **real** of real numbers—exact reals instead of true reals, though. Second, we do not require the presence of recursive types as SFPC, so as to simplify the presentation. Accommodating such type constructions is not central to our work, and can be handled through bilimit constructions [1, Section 5]. (Showing adequacy, as we will do in Section 10.2, in the presence of recursive types, is much more challenging, though. See [42] for adequacy in the presence of recursive types and discrete probabilistic choice, and [63] for a general technique for proving adequacy in the presence of recursive types.) Third, and finally, we choose to have an explicit type of distributions former D , in the style of Moggi’s Simple Metalanguage [60]: arrow types $\sigma \rightarrow \tau$ in SFPC are typically encoded as $\sigma \rightarrow D\tau$ in ISPCF.

Just like SFPC, ISPCF is a typed higher-order functional language. Its terms are given by the following grammar:

$$\begin{aligned} M, N, P, \dots ::= & x_{\sigma} \mid \underline{f} \mid \mathbf{sample}[\mu] \mid \mathbf{score}M \\ & \mid \lambda x.M \mid MN \mid \mathbf{rec} M \mid \mathbf{ret} M \mid \mathbf{do} x \leftarrow M; N \\ & \mid \langle M, N \rangle \mid \pi_1 M \mid \pi_2 M \mid \iota_1 M \mid \iota_2 M \mid \mathbf{case}MNP \end{aligned}$$

where x_{σ} (and also y_{τ}, z_{ν} , etc.) range over variables, whose type appears explicitly as a subscript (which we will often omit for clarity), \underline{f} ranges over a fixed set Σ of constants, and μ ranges over a fixed set \mathcal{M} of measures on \mathbb{R} . We assume a countably infinite supply of variables of each type.

We assume a function typ from Σ to the set of ISPCF types: $typ(\underline{f})$ is the type of \underline{f} , by definition.

Amongst those we require a constant \ast of type **unit**, and constants \underline{n} , one for each integer $n \in \mathbb{Z}$, with $typ(\underline{n}) = \mathbf{int}$.

$$\begin{array}{c}
\frac{}{\vdash x_\sigma : \sigma} \qquad \frac{\underline{f} \in \Sigma}{\vdash \underline{f} : \text{typ}(\underline{f})} \qquad \frac{\vdash M : \sigma \rightarrow \sigma}{\vdash \mathbf{rec} M : \sigma} \\
\\
\frac{}{\vdash \mathbf{sample}[\mu] : D\mathbf{real}} \qquad \frac{\vdash M : \mathbf{real}}{\vdash \mathbf{score} M : D\mathbf{unit}} \\
\\
\frac{\vdash M : \tau}{\vdash \lambda x_\sigma. M : \sigma \rightarrow \tau} \qquad \frac{\vdash M : \sigma \rightarrow \tau \quad \vdash N : \sigma}{\vdash MN : \tau} \\
\\
\frac{\vdash M : \tau}{\vdash \mathbf{ret} M : D\tau} \qquad \frac{\vdash M : D\sigma \quad \vdash N : D\tau}{\vdash \mathbf{do} x_\sigma \leftarrow M; N : D\tau} \\
\\
\frac{\vdash M : \sigma \quad \vdash N : \tau}{\vdash \langle M, N \rangle : \sigma \times \tau} \qquad \frac{\vdash M : \sigma \times \tau}{\vdash \pi_1 M : \sigma} \quad \frac{\vdash M : \sigma \times \tau}{\vdash \pi_2 M : \tau} \\
\\
\frac{\vdash M : \sigma}{\vdash \iota_1 M : \sigma + \tau} \quad \frac{\vdash M : \tau}{\vdash \iota_2 M : \sigma + \tau} \quad \frac{\vdash M : \sigma + \tau \quad \vdash N : \sigma \rightarrow v \quad \vdash P : \tau \rightarrow v}{\vdash \mathbf{case} MNP : v}
\end{array}$$

Fig. 1. The typing rules of ISPCF

We may include constants \underline{q} , one for each rational number q , with $\text{typ}(\underline{q}) = \mathbf{real}$. We may also include constants for non-rational numbers, such as $\underline{\pi}$; and standard numerical primitives such as $\underline{\sin}$, $\underline{\exp}$ or $\underline{\log}$, with $\text{typ}(\underline{\sin}) = \text{typ}(\underline{\exp}) = \text{typ}(\underline{\log}) = \mathbf{real} \rightarrow \mathbf{real}$, meant to implement the sine, exponential and logarithm functions respectively, as in SFPC.

We will see later that we can define a type \mathbf{bool} of Booleans, namely $\mathbf{void} + \mathbf{void}$, and two terms $\underline{\text{true}}$ and $\underline{\text{false}}$ of type \mathbf{bool} . Given that, we may include a sign test $\underline{\text{pos}} : \mathbf{real} \rightarrow \mathbf{bool}$, with the intention that $\underline{\text{pos}} M$ evaluates to the value $\underline{\text{true}}$ if M evaluates to a positive real number, and to the value $\underline{\text{false}}$ if M evaluates to a negative real number (we will discuss semantics soon).

The constructs \mathbf{ret} and \mathbf{do} are standard monadic constructs; \mathbf{rec} is a fixed-point combinator, and we assume the usual α -renaming conventions. Crucially, we keep the two iconic primitives of SFPC: \mathbf{sample} (or rather, $\mathbf{sample}[\mu]$) and \mathbf{score} . The typing rules are given in Figure 1.

We also write $M; N$ for $\mathbf{do} x \leftarrow M; N$, where x is a fresh variable—one that is not free in N .

While SFPC is call-by-value, ISPCF is a call-by-name language, just like Hakaru [72] or PPCF [17]. (Hakaru, however, lacks higher-order functions and recursion.) This slightly simplifies the denotational semantics, and does not preclude us from using a call-by-value style of programming if we so desire. Namely, call-by-value application $M@N$ of $M : \sigma \rightarrow D\tau$ to $N : D\sigma$ is defined as $\mathbf{do} x \leftarrow N; Mx$, where x is a fresh variable. We will return to this point in Section A.1.

8 THE DENOTATIONAL SEMANTICS OF ISPCF

The semantics $\llbracket \tau \rrbracket$ of types τ is given by induction on τ :

$$\begin{aligned}
\llbracket \mathbf{unit} \rrbracket &\stackrel{\text{def}}{=} \{*, \perp\} & \llbracket \mathbf{void} \rrbracket &\stackrel{\text{def}}{=} \{\perp\} & \llbracket \mathbf{int} \rrbracket &\stackrel{\text{def}}{=} \mathbb{Z}_\perp & \llbracket \mathbf{real} \rrbracket &\stackrel{\text{def}}{=} \mathbb{R}_\perp \\
\llbracket \sigma \rightarrow \tau \rrbracket &\stackrel{\text{def}}{=} \llbracket \llbracket \sigma \rrbracket \rightarrow \llbracket \tau \rrbracket \rrbracket & \llbracket D\tau \rrbracket &\stackrel{\text{def}}{=} \mathbf{V}_m(\llbracket \tau \rrbracket) \\
\llbracket \sigma \times \tau \rrbracket &\stackrel{\text{def}}{=} \llbracket \llbracket \sigma \rrbracket \times \llbracket \tau \rrbracket \rrbracket & \llbracket \sigma + \tau \rrbracket &\stackrel{\text{def}}{=} (\llbracket \llbracket \sigma \rrbracket \rrbracket + \llbracket \llbracket \tau \rrbracket \rrbracket)_\perp.
\end{aligned}$$

$$\begin{array}{ll}
\llbracket x_\sigma \rrbracket \rho \stackrel{\text{def}}{=} \rho(x_\sigma) & \llbracket f \rrbracket \rho \stackrel{\text{def}}{=} f \\
\llbracket \mathbf{sample}[\mu] \rrbracket \rho \stackrel{\text{def}}{=} i[\mu] & \llbracket \mathbf{score} M \rrbracket \rho \stackrel{\text{def}}{=} | \llbracket M \rrbracket \rho | \cdot \delta_* \\
\llbracket \lambda x_\sigma . M \rrbracket \rho \stackrel{\text{def}}{=} \lambda V \in \llbracket \sigma \rrbracket . \llbracket M \rrbracket (\rho[x_\sigma \mapsto V]) & \\
\llbracket MN \rrbracket \rho \stackrel{\text{def}}{=} \llbracket M \rrbracket \rho(\llbracket N \rrbracket \rho) & \llbracket \mathbf{rec} M \rrbracket \rho \stackrel{\text{def}}{=} \text{lfp}(\llbracket M \rrbracket \rho) \\
\llbracket \mathbf{ret} M \rrbracket \rho \stackrel{\text{def}}{=} \delta_{\llbracket M \rrbracket \rho} & \llbracket \mathbf{do} x_\sigma \leftarrow M; N \rrbracket \rho \stackrel{\text{def}}{=} (\llbracket \lambda x_\sigma . N \rrbracket \rho)^\dagger(\llbracket M \rrbracket \rho) \\
\llbracket \langle M, N \rangle \rrbracket \rho \stackrel{\text{def}}{=} (\llbracket M \rrbracket \rho, \llbracket N \rrbracket \rho) & \llbracket \pi_i M \rrbracket \rho \stackrel{\text{def}}{=} \lambda(V_1, V_2). V_i \\
\llbracket \mathbf{case} MNP \rrbracket \rho \stackrel{\text{def}}{=} \begin{cases} \llbracket N \rrbracket \rho(a) & \text{if } \llbracket M \rrbracket \rho = (1, a) \\ \llbracket P \rrbracket \rho(b) & \text{if } \llbracket M \rrbracket \rho = (2, b) \\ \perp & \text{otherwise} \end{cases} & \llbracket i_i M \rrbracket \rho \stackrel{\text{def}}{=} (i, \llbracket M \rrbracket \rho)
\end{array}$$

Fig. 2. The semantics of ISPCF

In $\llbracket \mathbf{unit} \rrbracket$, we have $\perp < *$. \mathbb{Z}_\perp is \mathbb{Z} plus a fresh element \perp , ordered so that \perp is least and all integers are pairwise incomparable. $[X \rightarrow Y]$ denotes the dcpo of Scott-continuous maps from the dcpo X to the dcpo Y , ordered pointwise. Given any two pointed dcpos X and Y , $X \times Y$ is their ordinary product, ordered pointwise. $X + Y$ is their coproduct, and consists of elements $(1, x)$ with $x \in X$, $(2, y)$ with $y \in Y$, so that $(1, x) \leq (1, x')$ if and only if $x \leq x'$, $(2, y) \leq (2, y')$ if and only if $y \leq y'$, and $(1, x)$ and $(2, y)$ are incomparable.

The denotational semantics $\llbracket M \rrbracket \rho$ of each term M , in the environment ρ , is given in Figure 2. Environments map variables x_σ to elements of the corresponding dcpo $\llbracket \sigma \rrbracket$. The notation $\rho[x \mapsto V]$ stands for the environment that maps x to V and every other variable y to $\rho(y)$. The least fixed point operator lfp on every pointed dcpo X is such that $\text{lfp}(f) \stackrel{\text{def}}{=} \sup_{n \in \mathbb{N}} f^n(\perp)$ for every $f \in [X \rightarrow X]$. In the denotation of $\mathbf{case} MNP$, $\llbracket N \rrbracket \rho(a)$ is the application of $\llbracket N \rrbracket \rho$, which is a function, to the value a .

Let us make a few comments. We assume an element $f \in \llbracket \tau \rrbracket$ for each constant $\underline{f} \in \Sigma$ of type τ . Hence, for example, we have elements $\pi \in \llbracket \mathbf{real} \rrbracket$, $\sin, \exp, \log \in \llbracket \mathbf{real} \rightarrow \mathbf{real} \rrbracket$, etc. The case of sign tests $\underline{\text{pos}}$ is interesting: by an easy argument based on the fact that the real line is connected, there is no Scott-continuous function from \mathbb{R}_\perp to $\llbracket \mathbf{bool} \rrbracket$, where $\mathbf{bool} \stackrel{\text{def}}{=} \mathbf{unit} + \mathbf{unit}$, which would map the elements $[a, a]$ with $a \geq 0$ to $\text{true} \stackrel{\text{def}}{=} (1, *)$, and those with $a < 0$ to $\text{false} \stackrel{\text{def}}{=} (2, *)$ (see [14, Proposition 2.4] for a generalization of this observation). And indeed, such tests are not computable. One reasonable choice for the semantics of $\underline{\text{pos}}$, if we decide to include it, is to define $\text{pos}([a, b])$ as true if $a > 0$, false if $b < 0$, and \perp otherwise.

The semantics of $\mathbf{sample}[\mu]$, namely $i[\mu]$, is the minimal valuation associated with the measure μ on \mathbb{R} , see Theorem 6.4.

For every $\mathbf{a} \in \mathbb{R}_\perp$, the notation $|\mathbf{a}|$ denotes 0 if $\mathbf{a} = \perp$ or if $\mathbf{a} = [a, b]$ with $a \leq 0 \leq b$, a if $\mathbf{a} = [a, b]$ with $a \geq 0$, and $-b$ if $\mathbf{a} = [a, b]$ with $b \leq 0$.

This, as well as the semantics of $\underline{\text{pos}}$, is an instance of a more general, well-known domain-theoretic construction. A *bc-domain* is a continuous dcpo in which every subset with an upper bound has a least upper bound. The bc-domains with their Scott topology are exactly the *densely injective* T_0 topological spaces, namely the T_0 spaces Z such that, for every dense subset X of any space Y , every continuous map $f: X \rightarrow Z$ extends to a continuous map from the whole of Y to Z [27, Proposition II-3.11]. There is even a pointwise largest such extension f^* , defined

by $f^*(y) \stackrel{\text{def}}{=} \sup_{U \in \mathcal{O}Y, y \in U} \inf_{x \in U \cap X} f(x)$ [27, Proposition II-3.9]. The latter formula makes sense and defines a continuous map f^* even when f is not continuous, but in that case f^* will fail to extend f .

One can check that $\llbracket \mathbf{unit} \rrbracket$, $\llbracket \mathbf{void} \rrbracket$, $\llbracket \mathbf{int} \rrbracket$, $\llbracket \mathbf{real} \rrbracket$, and in fact $\llbracket \tau \rrbracket$ for any type τ that does not contain the D operator, are bc-domains. In particular, every continuous map $g: \mathbb{R} \rightarrow \mathbb{R}$ induces a continuous map $\hat{g} \stackrel{\text{def}}{=} (i \circ g)^*: \mathbb{IR}_\perp \rightarrow \mathbb{IR}_\perp$ that extends $i \circ g$, namely: $\hat{g}([x, x]) = [g(x), g(x)]$ for every $x \in \mathbb{R}$. In doing so, we silently equate \mathbb{R} with a subspace of \mathbb{IR}_\perp through i , and it is easy to realize that \mathbb{R} is indeed dense in \mathbb{IR}_\perp .

This applies to \sin , \cos , \exp , notably. The case of \log is slightly different, since it is only defined on $\mathbb{R}_+ \setminus \{0\}$. In that case, the map $f: \mathbb{R} \rightarrow \mathbb{IR}_\perp$ that sends every positive real number x to $i(\log x) = [\log x, \log x]$, and all other numbers to \perp , is continuous, and it is natural to define $\log: \mathbb{IR}_\perp \rightarrow \mathbb{IR}_\perp$ as its largest continuous extension. The notation $|a|$, as used in the semantics of **score** M (where a is $\llbracket M \rrbracket \rho$), is equal to f^* where f is the (continuous) absolute value map from \mathbb{R} to $\overline{\mathbb{R}}_+$. Here $\overline{\mathbb{R}}_+$ is equipped with the Scott topology and is obviously a bc-domain, and therefore f extends to \mathbb{IR}_\perp . The semantics pos of pos is f^* , where f is the non-continuous map that sends every positive number to true and all other numbers to false. Hence f^* does not extend f , but we still have $f^*([a, a]) = f(a)$ if $a \neq 0$.

It is clear that for each derivable judgment $\vdash M: \tau$, for every environment ρ , $\llbracket M \rrbracket \rho$ is a well-defined element of $\llbracket \tau \rrbracket$. In particular, the least fixed point $\text{lfp}(\llbracket M \rrbracket \rho)$ is well-defined in the definition of $\llbracket \mathbf{rec} M \rrbracket \rho$, because $\llbracket \tau \rrbracket$ is a pointed dcpo; this is shown by induction on τ , and the least element of $\llbracket D\tau \rrbracket$ is the constant zero valuation. We also mention the following case, which shows where tensorial strengths are required. There is a morphism:

$$TX \times [X \rightarrow TY] \xrightarrow{t'_{X,Y}} T(X \times [X \rightarrow TY]) \xrightarrow{\text{App}^\dagger} TY$$

where $X \stackrel{\text{def}}{=} \llbracket \sigma \rrbracket$, $Y \stackrel{\text{def}}{=} \llbracket \tau \rrbracket$, T is the \mathbf{V}_m monad, t' is its dual tensorial strength, and App is application. Applying this to $(\llbracket M \rrbracket \rho, \llbracket \lambda x_\sigma. N \rrbracket \rho)$ yields the semantic value $\llbracket \mathbf{do} x_\sigma \leftarrow M; N \rrbracket \rho$.

Remark 8.1. By standard category-theoretic arguments, since \mathbf{V}_m is commutative, the following equation holds when x is not free in N and y is not free in M :

$$\llbracket \mathbf{do} x \leftarrow M; \mathbf{do} y \leftarrow N; P \rrbracket \rho = \llbracket \mathbf{do} y \leftarrow N; \mathbf{do} x \leftarrow M; P \rrbracket \rho.$$

It was already argued [81, Comments after Corollary 6.11] that this semantic equation is crucial in program transformation techniques such as disintegration-based Bayesian inference [72] as implemented in the Hakaru system [61]. As in [81], this equation is valid at all types in our semantics.

9 EXAMPLES

Let us give a few examples, not for their own sake, but in order to demonstrate that our approach is capable of defining a rich set of distributions, and also to show how one can reason on the semantics of ISPCF programs. Additional examples are given in Appendix A.1.

9.1 Rejection sampling

Let us imagine that you have a probability measure μ on a space X , let A be a measurable subset of X such that $\mu(A) \neq 0$. Then there is a conditional probability measure μ_A , defined by $\mu_A(E) \stackrel{\text{def}}{=} \mu(E)/\mu(A)$ for every measurable subset E of A . Rejection sampling allows one to draw an element at random in A with probability μ_A , by drawing an element in X with probability μ until we find one in A . This terminates with probability 1, and in fact in $1/\mu(A)$ trials on average.

Let τ be any type. We use **ifMNP** as an abbreviation for **caseMNP** when M has type **bool**, and we recall that **bool = void + void**. We consider the following ISPCF term:

$$\text{rej} \stackrel{\text{def}}{=} \lambda p_{D\tau}. \lambda \text{sel}_{\tau \rightarrow \text{bool}}. \mathbf{rec}(\lambda r_{D\tau}. \mathbf{do} \ x_{\tau} \leftarrow p; \mathbf{if} \ (\text{sel} \ x) \ (\mathbf{ret} \ x) \ r).$$

Intuitively, rej takes a distribution p as input, a selection function sel , and draws x at random with probability p until $\text{sel} \ x$ returns true. Let us examine the semantics of rej a bit more formally.

For every environment ρ , $\llbracket \text{rej} \rrbracket \rho$ is a function that takes a minimal valuation μ (bound to the variable p) and a selection function $s \in \llbracket \tau \rightarrow \text{bool} \rrbracket$ (bound to sel) to a minimal valuation $v_{|\text{sel}}$, which we now elucidate. By definition, $v_{|\text{sel}}$ is the least minimal valuation such that $v_{|\text{sel}} = \Phi(v_{|\text{sel}})$, where:

$$\Phi(v) = \llbracket \mathbf{do} \ x \leftarrow p; \mathbf{if} \ (\text{sel} \ x) \ (\mathbf{ret} \ x) \ r \rrbracket \rho [p \mapsto \mu, \text{sel} \mapsto s, r \mapsto v]$$

Let $U_1 \stackrel{\text{def}}{=} s^{-1}(\{\text{true}, (1, \perp)\})$ and $U_2 \stackrel{\text{def}}{=} s^{-1}(\{\text{false}, (2, \perp)\})$. Expanding the semantics shows that $\Phi(v)$ is the continuous valuation such that, for every $U \in \mathcal{O}[\llbracket \tau \rrbracket]$,

$$\begin{aligned} \Phi(v)(U) &= \int_{x \in \llbracket \tau \rrbracket} (\chi_{U_1}(x) \cdot \chi_U(x) + \chi_{U_2}(x) \cdot v(U)) \, d\mu \\ &= \mu(U_1 \cap U) + \mu(U_2) \cdot v(U). \end{aligned}$$

In other words, $\Phi(v) = \mu_{|U_1} + \mu(U_2) \cdot v$, where the *restriction* $\mu_{|U_1}$ is the continuous valuation defined by $\mu_{|U_1}(U) \stackrel{\text{def}}{=} \mu(U_1 \cap U)$ for every open set U , as introduced by Heckmann [37]; $\mu_{|U_1}$ is a minimal valuation since it is equal to $\Phi(0)$. In order to obtain the semantics of rej , let us solve the equation $v = \Phi(v)$ for v . One may compute the least solution as $\sup_{n \in \mathbb{N}} \Phi^n(0)$, where 0 denotes the zero valuation, but it is easier to solve the equation directly.

If $\mu(U_2) < 1$, then there are in general several solutions to the equation $v = \Phi(v)$. The smallest one is:

$$\frac{1}{1 - \mu(U_2)} \mu_{|U_1},$$

and this is the value of $\llbracket \text{rej} \rrbracket \rho(\mu)(s)$. When $0 < \mu(U_2) < 1$, there is another solution, and that is the continuous valuation that maps every non-empty open subset of $\llbracket \tau \rrbracket$ to $+\infty$, the *largest* continuous valuation on $\llbracket \tau \rrbracket$. (Yes, that is a minimal valuation, since it is simply the directed supremum of all simple valuations whatsoever.)

If $\mu(U_2) = 1$, then we are typically in the case of sampling with respect to a set of measure zero. The situation is more complicated in general, since $\mu(U_1) + \mu(U_2)$ may not be equal to 1: the condition $\mu(U_2) = 1$ imposes no constraint on $\mu_{|U_1}$.

The general case where $\mu(U_2) \geq 1$ is probably even less intuitive. The largest continuous valuation is always a solution to the equation $v = \Phi(v)$, but there is smaller one. For every open subset U of $\llbracket \tau \rrbracket$, the possible values r for $v(U)$ satisfy $r = \mu_{|U_1}(U) + \mu(U_2) \cdot r$: if $\mu_{|U_1}(U) = 0$, then the least value for r is 0 (this is the only possible value, apart from $+\infty$, if $\mu(U_2) > 1$; if $\mu(U_2) = 1$, every element of $\overline{\mathbb{R}}_+$ is a possible value); otherwise, the only possible value for r is $+\infty$. This suggests that the least solution v to $v = \Phi(v)$ is defined by:

$$v(U) = \begin{cases} 0 & \text{if } \mu_{|U_1}(U) = 0 \\ +\infty & \text{otherwise.} \end{cases}$$

This is just the continuous valuation $(+\infty) \cdot \mu_{|U_1}$. It is a minimal valuation since $\mu_{|U_1}$ is, and **because minimal valuations are closed under scalar multiplication**. Hence the desired semantics $\llbracket \text{rej} \rrbracket \rho$ is $(+\infty) \cdot \mu_{|U_1}$ if $\mu(U_2) \geq 1$.

9.2 Generating normal distributions

Let us imagine that we do not have a term `sample` $[\mathcal{N}(0, 1)]$ for the normal (Gaussian) distribution $\mathcal{N}(0, 1)$ with mean 0 and standard deviation 1. We will give two possible implementations in ISPCF here, as well as several implementations of some other well-known algorithms in Appendix 9.2.

For every measurable map $f: \mathbb{R} \rightarrow \overline{\mathbb{R}}_+$, $\mathcal{N}(0, 1)$ is such that:

$$\int_{x \in \mathbb{R}} f(x) d\mathcal{N}(0, 1) = \int_{x \in \mathbb{R}} f(x) \sqrt{\frac{1}{2\pi}} e^{-\frac{1}{2}x^2} dx.$$

This is a measure defined from Lebesgue measure λ on \mathbb{R} by applying the density map $x \mapsto \sqrt{\frac{1}{2\pi}} e^{-\frac{1}{2}x^2}$. As such, we can implement it (or rather, the corresponding valuation, obtained by restriction to the open sets; as usual, since $\mathcal{N}(0, 1)$ is locally finite, this does not cause any ambiguity) as follows, assuming a term `sample` $[\lambda]$, as well as primitives for multiplication, division, exponential `exp`, square root `sqrt`, and various numerical constants, all with their intended semantics. We agree that product is written with an infix \times , and we remember that $M; N$ abbreviates `do x ← M; N`, where x is a fresh variable.

$$\text{normal} \stackrel{\text{def}}{=} \text{do } x_{\text{real}} \leftarrow \text{sample}[\lambda]; \text{score}(\text{sqrt}(0.5/\pi) \times \text{exp}(-0.5 \times x \times x)); \text{ret } x.$$

This is the original purpose of `score`: to define a distribution from another one through a density function. We check that this gives us the intended result. For every environment ρ , for every open subset W of \mathbb{IR}_{\perp} ,

$$\llbracket \text{normal} \rrbracket \rho(W) = \int_{x \in \mathbb{IR}_{\perp}} |\bar{g}(x)| \cdot \delta_x(W) di[\lambda]$$

where $\bar{g}(x) \stackrel{\text{def}}{=} \llbracket \text{sqrt}(0.5/\pi) \times \text{exp}(-0.5 \times x \times x) \rrbracket \rho[x \mapsto x]$; we check that $\bar{g}([x, x]) = [g(x), g(x)]$ where $g(x) \stackrel{\text{def}}{=} \sqrt{\frac{1}{2\pi}} e^{-\frac{1}{2}x^2}$. By the change-of-variables formula, and the fact that $\delta_{[x, x]}(W) = \chi_{i^{-1}(W)}(x)$,

$$\llbracket \text{normal} \rrbracket \rho(W) = \int_{x \in \mathbb{R}} g(x) \cdot \chi_{i^{-1}(W)}(x) d\lambda = \mathcal{N}(0, 1)(i^{-1}(W)).$$

Therefore $\llbracket \text{normal} \rrbracket \rho = i[\mathcal{N}(0, 1)]$.

Another way of implementing $\mathcal{N}(0, 1)$ is the *Box-Muller algorithm* [9], which produces pairs of two independent normally distributed variables, from two independent uniformly distributed variables in $[0, 1]$. We should say that one of our purposes in explaining its implementation in ISPCF is to dispel the myth that a semantics based on continuous maps, such as ours, would necessarily fail to be expressive enough to encode measures with non-compact support using only measures with compact support. We also give its formal semantics.

In order to implement the Box-Muller algorithm in ISPCF, we assume a term `sample` $[\lambda_1]$, where λ_1 is the uniform measure on $[0, 1]$ (a measure with compact support). We also assume we have primitives `sin`, `cos`, `log`, and `sqrt`, as well as addition, subtraction, opposite, and multiplication, with the expected semantics. We consider the following ISPCF term of type $D(\text{real} \times \text{real})$. We use the shorthand `let x = u in v` for $(\lambda x.v)u$.

$$\begin{aligned} \text{box_muller} &\stackrel{\text{def}}{=} \text{do } x_{\text{real}} \leftarrow \text{sample}[\lambda_1]; \text{do } y_{\text{real}} \leftarrow \text{sample}[\lambda_1]; \\ &\quad \text{let } c_{\text{real}} = \text{cos}(2.0 \times \pi \times y) \text{ in let } s_{\text{real}} = \text{sin}(2.0 \times \pi \times y) \text{ in} \\ &\quad \text{let } m_{\text{real}} = \text{sqrt}(-2.0 \times \log x) \text{ in} \\ &\quad \text{ret}((m \times c, m \times s)). \end{aligned}$$

This is a non-recursive definition, and as a consequence, $\llbracket \text{box_muller} \rrbracket \rho$ is a probability distribution on $\mathbb{IR}_\perp \times \mathbb{IR}_\perp$ that is actually concentrated on the subspace $\mathbb{R} \times \mathbb{R}$. Explicitly, for every open subset W of $\mathbb{IR}_\perp \times \mathbb{IR}_\perp$,

$$\llbracket \text{box_muller} \rrbracket \rho(W) = \int_{\mathbf{x} \in \mathbb{IR}_\perp} \left(\int_{\mathbf{y} \in \mathbb{IR}_\perp} \delta_{f(\mathbf{x}, \mathbf{y})}(W) \, di[\lambda_1] \right) di[\lambda_1],$$

where $f(\mathbf{x}, \mathbf{y})$ is $\llbracket \text{let } c_{\text{real}} = \underline{\text{cos}}(2.0 \times \underline{\pi} \times y) \text{ in } \dots \text{ in } \langle m \times c, m \times s \rangle \rrbracket \rho[x \mapsto \mathbf{x}, y \mapsto \mathbf{y}]$. By the change-of-variables formula, we have:

$$\llbracket \text{box_muller} \rrbracket \rho(W) = \int_{x \in \mathbb{R}} \left(\int_{y \in \mathbb{R}} \delta_{f([x, x], [y, y])}(W) \, d\lambda_1 \right) d\lambda_1,$$

and $f([x, x], [y, y]) = [g(x, y), g(x, y)]$, where $g(x, y) = (\sqrt{-2 \log x} \cos(2\pi y), \sqrt{-2 \log x} \sin(2\pi y))$. It follows that $f([x, x], [y, y]) \in W$ if and only if $g(x, y) \in i^{-1}(W)$, so that:

$$\llbracket \text{box_muller} \rrbracket \rho(W) = \int_{x \in \mathbb{R}} \left(\int_{y \in \mathbb{R}} \chi_{i^{-1}(W)}(g(x, y)) \, d\lambda_1 \right) d\lambda_1,$$

We use the familiar argument subtending the Box-Muller algorithm and obtain that for every measurable map $h: \mathbb{R}^2 \rightarrow \overline{\mathbb{R}}_+$ (the integrals are ordinary Lebesgue integrals),

$$\int_{(s, t) \in \mathbb{R}^2} h(s, t) \, d\mathcal{N}(0, 1) \otimes \mathcal{N}(0, 1) = \int_{x, y \in [0, 1]} h(\sqrt{-2 \log x} \cos(2\pi y), \sqrt{-2 \log x} \sin(2\pi y)) \, dx \, dy.$$

By taking $h \stackrel{\text{def}}{=} \chi_{i^{-1}(W)}$, it follows that $(\mathcal{N}(0, 1) \otimes \mathcal{N}(0, 1))(i^{-1}(W)) = \llbracket \text{box_muller} \rrbracket \rho(W)$. Therefore box_muller implements the product $\mathcal{N}(0, 1) \otimes \mathcal{N}(0, 1)$, in the sense that:

$$\llbracket \text{box_muller} \rrbracket \rho = i[\mathcal{N}(0, 1) \otimes \mathcal{N}(0, 1)].$$

9.3 Generating Lebesgue valuation on \mathbb{R}

We have noticed that one can define **continuous valuations** with unbounded support from just **sample** $[\lambda_1]$, although λ_1 has bounded support. We now show that we can define **continuous valuations** with unbounded support and whose total mass is infinite as well. The method is known as *importance sampling*, in the sense of [76] for example, and requires the use of **score**. What may be intriguing is that **score** multiplies the current ‘probability’ by a real number different from $+\infty$. Still, this can be used to increase the total mass from 1 to $+\infty$.

Hence, we imagine that the term **sample** $[\lambda_1]$ is available, where λ_1 is Lebesgue measure on $[0, 1]$, but that **sample** $[\lambda]$ is not, where λ is Lebesgue measure on \mathbb{R} . Assuming constants $\underline{\pi}$, $\underline{1.0}$, $\underline{2.0}$, $\underline{\tan}$, $\underline{\arctan}$, and some constants for addition, subtraction, multiplication and division, we define:

$$\begin{aligned} \text{uniform} &\stackrel{\text{def}}{=} \lambda a_{\text{real}} . \lambda b_{\text{real}} . \text{do } x_{\text{real}} \leftarrow \text{sample}[\lambda_1]; \text{ret } (a + (b - a) \times x) \\ \text{lebesgue} &\stackrel{\text{def}}{=} \text{do } z_{\text{real}} \leftarrow \text{uniform}(-\underline{\pi}/\underline{2.0})(\underline{\pi}/\underline{2.0}); \text{score}(\underline{1.0}/(\underline{1.0} + z \times z)); \text{ret}(\underline{\arctan} z). \end{aligned}$$

We claim that **lebesgue** fits the bill. Let $v_{[a, b]}$ denote the uniform probability measure on the interval $[a, b]$; up to the embedding $i: \mathbb{R} \rightarrow \mathbb{IR}_\perp$, this is what **uniform** applied to a and b computes. Then we use a monotonic, differentiable homeomorphism g from \mathbb{R} to the open interval $]a, b[$ in order to transport that to the whole of \mathbb{R} . (One may observe that $]a, b[$ differs from $[a, b]$, but that will not matter, since the complement of $]a, b[$ in $[a, b]$ has $v_{[a, b]}$ -measure 0.) We have chosen $g(u) \stackrel{\text{def}}{=} \arctan u$, $a = -\pi/2$, $b = \pi/2$ here. The formal verification of this computation, namely of the fact that $\llbracket \text{lebesgue} \rrbracket \rho = i[\lambda]$, is left to the reader; see Appendix A.2 for details.

9.4 Generating exponential distributions

Generating an exponential distribution is another classical example. The novelty is that this is defined from Lebesgue measure λ by a density function that is *not* continuous. Although ISPCF can only express continuous functions, we show that this will not prevent us from defining the exponential distribution.

For simplicity, we consider the exponential distribution with parameter 1. The density map is:

$$g(x) \stackrel{\text{def}}{=} \begin{cases} e^{-x} & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

We note that g is not continuous at 0. However, we have the following candidate ISPCF term, assuming the relevant primitives:

$$\text{exp_density} \stackrel{\text{def}}{=} \lambda x_{\mathbf{real}}. \mathbf{if} (\text{pos } x) (\text{exp}(-x)) \underline{0}.0.$$

For every environment ρ , we see that $\bar{g} \stackrel{\text{def}}{=} \llbracket \text{exp_density} \rrbracket$ maps \perp to \perp , and every interval $[a, b]$ to $[e^{-b}, e^{-a}]$ if $a > 0$, to $[0, 0]$ if $b < 0$, and to \perp otherwise. In particular, $\llbracket \text{exp_density} \rrbracket (i(a))$ coincides with $i(g(a))$ for all points of continuity of g , namely for every $a \neq 0$.

The most obvious way to implement the exponential distribution is by using **score**. We assume a term **sample** $[\lambda]$, where λ is Lebesgue measure on \mathbb{R} .

$$\text{expo} \stackrel{\text{def}}{=} \mathbf{do} x_{\mathbf{real}} \leftarrow \mathbf{sample}[\lambda]; \mathbf{score}(\text{exp_density } x); \mathbf{ret} x.$$

For every open subset W of \mathbb{IR}_{\perp} , we compute:

$$\begin{aligned} \llbracket \text{expo} \rrbracket \rho(W) &= \int_{\mathbf{x} \in \mathbb{IR}_{\perp}} |\bar{g}(\mathbf{x})|. \delta_{\mathbf{x}}(W) \, di[\lambda] \\ &= \int_{x \in \mathbb{R}} |\bar{g}(i(x))|. \delta_{i(x)}(W) \, d\lambda && \text{by the change-of-variables formula} \\ &= \int_{x \in \mathbb{R}} g(x). \chi_{i^{-1}(W)}(x) \, d\lambda. \end{aligned}$$

The last line is justified by the fact that $\delta_{i(x)}(W) = \chi_{i^{-1}(W)}(x)$, and more importantly, by the fact that $|\bar{g}(i(x))|$ and $g(x)$ coincide for every $x \in \mathbb{R}$ except on a set of Lebesgue measure 0. (Namely, when $x = 0$, in which case $g(x) = 1$ and $|\bar{g}(i(x))| = 0$.)

It follows that $\llbracket \text{expo} \rrbracket \rho$ is exactly $i[g \cdot \lambda]$, the image by i of the exponential distribution $g \cdot \lambda$. Other implementations of the exponential distribution are described, and proved correct, in Appendix A.3.

9.5 Distributions on higher-order objects

All our previous examples were about building distributions on simple types such as **real** or **real** \times **real**. One may be interested in generating distributions on functions, or even on types of distributions themselves.

Let us start with the following problem. We are given a countably infinite family of distributions on some type τ , as the value of a parameter $F: \mathbf{int} \rightarrow D\tau$, and we wish to find a random function f of type $\mathbf{int} \rightarrow \tau$, such that $f(0), f(1), \dots$, are independently distributed according to the distributions $F(0), F(1), \dots$, respectively. Formally, we look for a distribution of type $D(\mathbf{int} \rightarrow \tau)$ over those functions.

At first sight, this looks like an infinite-dimensional generalization of the product distribution term $\text{prod}_{\sigma, \tau}$ introduced earlier. [Theorem 5.3 of \[34\]](#), which states that projective limits of diagrams of finite products of continuous valuations

always exist, may give the impression that the question is easy. But infinite products of continuous valuations, or of measures for that matter, do not exist in general. More to the point, the question we wish to address is whether, and if so how, we can *implement* them in ISPCF when they exist. In order to appreciate the difficulty, we start with a naive proposal for an implementation, and we explain why this first attempt is bound to fail; we will give an actual solution further down below. Here is this naive proposal:

$$\text{wrong_infinite_prod}_\tau \stackrel{\text{def}}{=} \text{rec}(\lambda P_{(\mathbf{int} \rightarrow D\tau) \rightarrow D(\mathbf{int} \rightarrow \tau)}. \lambda F_{\mathbf{int} \rightarrow D\tau}. \\ \text{do } x_\tau \leftarrow \text{head } F; \text{ do } \text{rest}_{\mathbf{int} \rightarrow \tau} \leftarrow P(\text{tail } F); \text{ret}(x :: \text{rest})),$$

where $\text{head}F \stackrel{\text{def}}{=} F(0)$ extracts the first element of the sequence encoded by F , $\text{tail}F \stackrel{\text{def}}{=} \lambda n_{\mathbf{int}}. F(n + 1)$ extracts the remaining elements, and $x :: \text{rest} \stackrel{\text{def}}{=} \lambda n_{\mathbf{int}}. \text{if}(n = 0) x (\text{rest } (n - 1))$ adds x to the front of rest . This may seem to fit the bill: it draws x at random along the distribution denoted by $F(0)$, then recurses in order to draw the remaining elements at indices 1, 2, ... However, and as the lack of base case for the recursion may hint of, this is wrong: the semantics of $\text{wrong_infinite_prod}_\tau$ is merely the zero valuation; that program never terminates.

In fact, our problem has no solution: **infinite products of continuous valuations (or of measures) do not exist in general**. The reason is that the total mass of the required infinite product is in general undefined. Indeed, let us imagine that we are given infinitely many distributions $\mu_0, \mu_1, \dots, \mu_n, \dots$ on the same space $X \stackrel{\text{def}}{=} \llbracket \tau \rrbracket$, as input. Their infinite product, if it exists, has total mass $\prod_{n=0}^{+\infty} a_n$, where $a_n \stackrel{\text{def}}{=} \mu_n(X)$ is the total mass of μ_n . That only makes sense if the infinite product $\prod_{n=0}^{+\infty} a_n$ converges. For example, it makes non sense if $a_n = 1/2$ for n odd and $a_n = 2$ for n even. Such cases are easy to build using **score**.

Let us consider a slightly easier problem, where all the distributions μ_n are constrained to be the same distribution μ ; we wish to compute the infinite product μ^∞ of μ with itself. Now this simplified problem still does not have any solution (unless $\llbracket \tau \rrbracket$ is empty), and this is now particular to our domain-theoretic semantics. Indeed, let us imagine that the map $f: \mu \mapsto \mu^\infty$ were computable, hence Scott-continuous, and let x be any fixed element of $\llbracket \tau \rrbracket$. Then the map $a \in [0, 1] \mapsto (a\delta_x)^\infty(\llbracket \tau \rrbracket)$ would be Scott-continuous. But that function maps 1 to 1, and all other elements of $[0, 1]$ to 0, and is therefore not Scott-continuous.

Instead, we will show how one can answer our problem in special cases. The first special case we consider is to build the product λ_1^∞ of a countably infinite number of copies of Lebesgue valuation λ_1 on $[0, 1]$. By definition, λ_1^∞ is the unique continuous valuation on the countable topological product $[0, 1]^\mathbb{N}$ whose image valuation $\pi_S[\lambda_1^\infty]$ onto $[0, 1]^S$ is the (finite) product valuation $\prod_{n \in S} \lambda_1$, for every finite subset S of \mathbb{N} . We write $\pi_S: [0, 1]^\mathbb{N} \rightarrow [0, 1]^S$ for projection onto the coordinates in S . This exists and is unique by general theorems [34, Theorem 5.3].

We embed $[0, 1]^\mathbb{N}$, and more generally $\mathbb{R}^\mathbb{N}$, into $\llbracket \mathbf{int} \rightarrow \mathbf{real} \rrbracket = [\mathbb{Z}_\perp \rightarrow \mathbb{IR}_\perp]$ as follows. There is a map $j: \mathbb{R}^\mathbb{N} \rightarrow \llbracket \mathbf{int} \rightarrow \mathbf{real} \rrbracket$ defined by $j(\vec{x})(n) \stackrel{\text{def}}{=} i(x_n)$ for every $\vec{x} \stackrel{\text{def}}{=} (x_n)_{n \in \mathbb{N}}$ in $[0, 1]^\mathbb{N}$ and every $n \in \mathbb{N}$, where i is the usual embedding of \mathbb{R} into \mathbb{IR}_\perp , restricted to $[0, 1]$; and $j(\vec{x})(n) \stackrel{\text{def}}{=} \perp$ for every $n \in \{-1, -2, \dots\} \cup \{\perp\}$. In order not to disrupt the flow of exposition, the proof of the following lemma, as well as all other proofs in this section, are relegated to Appendix A.4.

LEMMA 9.1. *The Scott topology of $\llbracket \mathbf{int} \rightarrow \mathbf{real} \rrbracket$ has a subbase of open sets of the form $[n \in V] \stackrel{\text{def}}{=} \{f \in [\mathbb{Z}_\perp \rightarrow \mathbb{IR}_\perp] \mid f(n) \in V\}$, where n ranges over \mathbb{Z}_\perp and V ranges over the open subsets of \mathbb{IR}_\perp .*

The map j is a topological embedding of $\mathbb{R}^\mathbb{N}$ into $\llbracket \mathbf{int} \rightarrow \mathbf{real} \rrbracket$.

We will build a term of type $D(\mathbf{int} \rightarrow \mathbf{real})$ whose semantics is $j[\lambda_1^\infty]$ —a continuous valuation that now makes sense; this will be stated in Theorem 9.6.

The idea is simple. We draw a real number at random using λ_1 , we write it in binary, cut (or dice) its sequence of bits into a countably infinite partition of subsequences of bits, and we reassemble (or splice) each subsequence into a new real number in $[0, 1]$. The dicing operation is far from being continuous, but this will not matter; the reason why will be given in Remark 9.1.

Let us introduce the mathematical objects we will need. Let ν be the uniform measure on infinite sequences of bits; this is the unique Borel measure on $\{0, 1\}^\mathbb{N}$ (where $\{0, 1\}$ has the discrete topology) such that $\nu(\{s \in \{0, 1\}^\mathbb{N} \mid s_0 = b_0, \dots, s_{k-1} = b_{k-1}\}) = 1/2^k$ for all $k \in \mathbb{N}$ and elements $b_0, \dots, b_{k-1} \in \{0, 1\}$. The easiest way to show its existence is as the image measure of Lebesgue measure by a suitable map.

LEMMA 9.2. *The function $\text{bin}: \mathbb{R} \rightarrow \{0, 1\}^\mathbb{N}$ that maps every number in $[0, 1[$ to the string of bits in its binary expansion (namely, the i th bit in $\text{bin}(x)$ is $\lfloor 2^i \cdot x \rfloor$, for any $i \geq 1$), every negative number to the all zero string 0^ω , and every number larger than or equal to 1 to the all one string 1^ω is measurable. The image measure $\text{bin}[\lambda_1]$ is the unique measure ν on $\{0, 1\}^\mathbb{N}$ such that $\nu(\{s \in \{0, 1\}^\mathbb{N} \mid s_0 = b_0, \dots, s_{k-1} = b_{k-1}\}) = 1/2^k$ for all $k \in \mathbb{N}$ and elements $b_0, \dots, b_{k-1} \in \{0, 1\}$.*

The function bin is not continuous on the whole of \mathbb{R} , hence not implementable. Its restriction to the set of non-dyadic numbers in $[0, 1]$, is continuous, though, and we make this clearer by defining a continuous map $\text{bin}' : \mathbb{R}_\perp \rightarrow \{0, 1\}_\perp^\mathbb{N}$ that implements bin on the non-dyadic numbers in $[0, 1]$. Roughly, bin' is implemented as follows. Given an argument value of type \mathbf{real} , if the value is smaller than $1/2$ (which one can test by using pos), then emit a zero bit and multiply the value by 2; if that value is larger than $1/2$, then emit a one bit, subtract $1/2$ from the value and multiply the result by 2; then collect all the emitted bits into a sequence, which may be infinite, or finite (if computation ever gets stuck). Formally, we define $\text{bin}'(\underline{a})_k$ by induction on k as follows. If $\underline{a} = \perp$, then $\text{bin}'(\underline{a})_k$ is \perp . If $\underline{a} = [a, b]$ and $k = 0$, then $\text{bin}'([a, b])_0$ is defined as 0 if $b < 1/2$, 1 if $a > 1/2$, and as \perp otherwise. If $\underline{a} = [a, b]$ and $k \geq 1$, then $\text{bin}'([a, b])_k$ is defined as $\text{bin}'(\underline{c})_{k-1}$ where \underline{c} is equal to $[2a, 2b]$ if $b < 1/2$, $[2a - 1, 2b - 1]$ if $a > 1/2$, and \perp otherwise. One checks easily that bin' is continuous.

Remark 9.1. Let E_0 be the collection of all points that are either dyadic or outside $[0, 1]$. This is a set of λ_1 -measure zero, because there are only countably many dyadic numbers and λ_1 is supported on $[0, 1]$. Since bin and bin' coincide outside E_0 , the image measures of λ_1 by each of those two functions is the same. (Formally, $j'[\text{bin}[\lambda_1]]$ and $\text{bin}'[j[\lambda_1]]$ coincide, for any environment ρ , and where j' is the obvious topological embedding $\{0, 1\}_\perp^\mathbb{N} \rightarrow \{0, 1\}^\mathbb{N}$.)

LEMMA 9.3. *The map $\text{num}: \{0, 1\}^\mathbb{N} \rightarrow \mathbb{R}$ that sends s to $\sum_{i \in \mathbb{N}} s_i / 2^{i+1}$ is continuous, and the image measure $\text{num}[\nu]$ is equal to λ_1 .*

We assume a bijective computable pairing map $\langle _, _ \rangle : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$. A practical one is the map that interleaves the two binary representations of the numbers in argument, namely $\langle \sum_{i \in \mathbb{N}} a_i 2^i, \sum_{j \in \mathbb{N}} b_j 2^j \rangle \stackrel{\text{def}}{=} \sum_{i \in \mathbb{N}} a_i 2^{2i} + \sum_{j \in \mathbb{N}} b_j 2^{2j+1}$.

Given a random string of bits s , the strings $s[m] \stackrel{\text{def}}{=} (s_{\langle m, n \rangle})_{n \in \mathbb{N}}$, $m \in \mathbb{N}$, are themselves random and independent. We express this by the following well-known lemma. In order to state it, for every $s \in \{0, 1\}^\mathbb{N}$, let $\uparrow s$ (“shift s ”) be the string such that $(\uparrow s)_{\langle m, n \rangle} = s_{\langle m+1, n \rangle}$ for all $m, n \in \mathbb{N}$. In other words, $(\uparrow s)[m] = s[m+1]$ for every $m \in \mathbb{N}$.

Computing $s[m]$ from s and m is easy, considering that pairing $\langle _, _ \rangle$ is computable.

The spaces $\{0, 1\}^\mathbb{N}$ and $(\{0, 1\}^\mathbb{N})^2$ are compact Hausdorff, and therefore locally compact and sober. **They are also second-countable, since \mathbb{N} is countable. As we have seen in Section 2.3, locally finite continuous valuations and locally**

finite Borel measures, in particular also bounded continuous valuations and bounded Borel measures, are in one-to-one correspondence on such spaces. We will therefore equate the two notions.

LEMMA 9.4. *The map $split: s \mapsto (s[0], \uparrow s)$ is a homeomorphism of $\{0, 1\}^{\mathbb{N}}$ onto $(\{0, 1\}^{\mathbb{N}})^2$. The image continuous valuation (resp., measure) $split[v]$ is the product valuation (resp., product measure) $v \otimes v$.*

COROLLARY 9.5. *For every $N \in \mathbb{N}$,*

- (1) *the map $split_N: s \mapsto (s[0], s[1], \dots, s[N-1], \uparrow^N s)$ is a homeomorphism of $\{0, 1\}^{\mathbb{N}}$ onto $(\{0, 1\}^{\mathbb{N}})^{N+1}$;*
- (2) *the image measure $split_N[v]$ is the product measure $\underbrace{v \otimes v \otimes \dots \otimes v}_{N+1}$;*
- (3) *the map $\omega_N: s \mapsto (s[0], s[1], \dots, s[N-1])$ is continuous from $\{0, 1\}^{\mathbb{N}}$ to $(\{0, 1\}^{\mathbb{N}})^N$, and $\omega_N[v]$ is the product measure $\underbrace{v \otimes v \otimes \dots \otimes v}_N$.*

PROOF. We show 1 and 2 by induction on N . This is obvious if $N = 0$, and is by Lemma 9.4 if $N = 1$. If $N \geq 2$, we note that $split_N = (\text{id}_{\mathbb{R}} \times split_{N-1}) \circ split$. Hence $split_N$ is a homeomorphism, using the induction hypothesis, and $split_N[v] = (\text{id}_{\mathbb{R}} \times split_{N-1})[split[v]] = (\text{id}_{\mathbb{R}} \times split_{N-1})[v \otimes v]$ (by Lemma 9.4) $= v \otimes split_{N-1}[v] = v \otimes \underbrace{v \otimes v \otimes \dots \otimes v}_N$, by induction hypothesis. Item 3 follows from 1 and 2 by composing with the appropriate projection map. \square

Remark 9.2. It is also true that the image measure $split[v]$ is the product measure of v by itself. This is a standard exercise in measure theory: any two probability measures that agree on a π -system generating the σ -algebra are equal [6, Theorem 3.3]; a π -system is a family of sets closed under finite intersections, and it suffices to apply this result to the π -system consisting of the sets $U_{I,J} \stackrel{\text{def}}{=} \{s \in \{0, 1\}^{\mathbb{N}} \mid \forall m \in I, s_m = 0 \text{ and } \forall n \in J, s_n = 1\}$, where (I, J) ranges over pairs of disjoint finite subsets of \mathbb{N} . A subtle point with that approach is that this product is a product of $\{0, 1\}^{\mathbb{N}}$ with itself, certainly, but in which category? In Lemma 9.4, the product is taken in **Top**. The measure-theoretic approach requires to work with the product taken in the category of measurable spaces and measurable maps. In general, those two products differ. The problem is similar to the issue discussed in Section 4. Fortunately, there is no such problem here, because the Borel σ -algebra of the topological product of two Polish spaces coincides with the product of the two σ -algebras.

We assume an ISPCF constant $\text{mux}: \mathbf{real} \rightarrow \mathbf{int} \rightarrow \mathbf{real}$ whose semantics mux satisfies the following: for every $x \in \mathbb{R}$, for every $m \in \mathbb{N}$,

$$\text{mux}(i(x))(m) = i(\text{num}'(\text{bin}'(x)[m])),$$

where bin' was introduced before Remark 9.1, and where num' is any continuous map from $\{0, 1\}_{\perp}^{\mathbb{N}}$ to \mathbb{R}_{\perp} that extends num , in the sense that $\text{num}'(s) = i(\text{num}(s))$ for every $s \in \{0, 1\}^{\mathbb{N}}$; for example,

$$\text{num}'(s) \stackrel{\text{def}}{=} \begin{cases} i(\text{num}(s)) & \text{if } s \in \{0, 1\}^{\mathbb{N}} \\ [x, x + 1/2^k] & \text{where } x \stackrel{\text{def}}{=} \sum_{i=0}^{k-1} s_i/2^{i+1} \\ & \text{if } s_0, \dots, s_{k-1} \neq \perp, s_k = \perp \end{cases} \quad (4)$$

We are now ready to implement an ISPCF term computing λ_1^{∞} :

$$\text{rand_uniform_seq} \stackrel{\text{def}}{=} \mathbf{do} \, r_{\mathbf{real}} \leftarrow \mathbf{sample}[\lambda_1]; \mathbf{ret}(\text{mux} \, r).$$

This is a term of type $D(\mathbf{int} \rightarrow \mathbf{real})$, and we claim that it draws an infinite sequence of independent, uniformly distributed real numbers in $[0, 1]$. This is a pretty tedious verification, using the lemmas and corollaries states earlier in this section. For a complete proof, see Appendix A.4.

THEOREM 9.6. *For every environment ρ , $\llbracket \text{rand_uniform_seq} \rrbracket \rho = j[\lambda_1^\infty]$.*

We give other examples of distributions on higher-order objects in Appendix A.5, based on this construction, including an implementation of the Dirichlet process, a process that draws distributions at random.

By now, we hope to have provided enough examples in order to convince the reader that one can write enough useful distributions in a language with a domain-theoretic semantics such as ISPCF. Let us proceed with matters of operational semantics.

10 OPERATIONAL SEMANTICS

The simplest possible operational semantics of ISPCF is one where **sample** $[\mu]$ draws actual real numbers at random, following the distribution μ . We follow similar other proposals [10, 17, 81] pretty closely. The main difference is that we will draw *exact* reals, namely elements of \mathbb{IR}_\perp , instead of true reals in \mathbb{R} , at random. This is really not much of a difference, since the probability that **sample** $[\mu]$ draws an exact real number that is not a true real number is zero.

In order to describe our operational semantics formally, we consider generalized ISPCF terms, with extra constants, one for each real number, and we consider these generalized ISPCF terms as configurations of an abstract machine. We then define a probabilistic transition relation on the space of configurations. In order to do so, we will need to topologize the space of configurations, much as previous proposals [10, 17, 81] defined a σ -algebra on similar spaces of configurations. The probabilistic transition relation will be a map from configurations to measures (rather, continuous valuations) on the space of configurations. That map will be continuous, and not just measurable.

Notionally, a *generalized ISPCF term* is a closed ISPCF term built on a set of constants Σ that contains exactly one constant \underline{a} for each $a \in \mathbb{R}$. We will equate the already existing constants $\underline{r}, r \in \mathbb{R}$, with $[\underline{r}, r]$.

The following alternate definition will be more formal, and will allow us to give a simple description of the topology we will put on generalized ISPCF terms. We fix a countable enumeration $x_1, x_2, \dots, x_n, \dots$ of *template* variables of type **real** once and for all, in such a way that we still have an infinite supply of variables of type **real** not in that list. A *template* is an ISPCF term M whose free variables, read from left to right, are x_1, \dots, x_k for some $k \in \mathbb{N}$, and which does not contain any constant of type **real**. In particular, the only free variables of M are template variables, they occur only once in M , and are numbered from 1 to k consecutively from left to right. We let $\text{fv}(M)$ be the set of free variables of a term M . Then we equate a generalized ISPCF term with a *configuration* (M, θ) where M is a template and $\theta \in \mathbb{IR}_\perp^{\text{fv}(M)}$. Replacing each template variable x_i that is free in M by $\theta(x_i)$ yields a generalized ISPCF term, which we write as $M\theta$. Conversely, each generalized ISPCF term is represented by a unique configuration (M, θ) : M has to be the given generalized ISPCF term where each constant \underline{r} of type **real** has been replaced by a fresh template variable, which is then mapped to r by θ . We call M the *shape* of the generalized ISPCF term. For example, the generalized ISPCF term $\lambda y. \underline{2.0} \times y + \underline{1.0}$ is represented by the configuration $(\lambda y. x_1 \times y + x_2, [x_1 \mapsto 2.0, x_2 \mapsto 1.0])$.

Definition 10.1 (Space of configurations Γ). Γ denotes the set of all configurations, ordered by $(M, \theta) \leq (N, \theta')$ if and only if $M = N$ and $\theta \leq \theta'$, namely, for every $x \in \text{fv}(M)$, $\theta(x) \leq \theta'(x)$.

Remark 10.1. Γ is a dcpo, and in fact a continuous dcpo. To see the latter, Γ is a coproduct over all possible shapes of finite products of copies of \mathbb{IR}_\perp . Each copy of \mathbb{IR}_\perp is a continuous dcpos, and continuous dcpos are closed under finite products and arbitrary coproducts.

This structure transports to a structure of continuous dcpo on the set of generalized ISPCF terms through the bijection $(M, \theta) \mapsto M\theta$.

While our algebra of terms was rather open-ended until now, we will need to restrict it slightly, and we will assume that Σ has *strictly observable first-order constants*, see below.

Definition 10.2 (Basic and observable types). The algebra of *observable types* is:

$$\beta ::= \mathbf{unit} \mid \mathbf{void} \mid \mathbf{int} \mid \mathbf{real} \mid \beta_1 + \beta_2 \mid \beta_1 \times \beta_2,$$

The *basic types* are **unit**, **void**, **int**, and **real**.

In other words, the observable types are all types that do not contain \rightarrow or D .

Definition 10.3 (Observable element). The *observable elements* of $\llbracket \beta \rrbracket$, where β is an observable type, are defined as follows. The observable elements of $\llbracket \mathbf{void} \rrbracket$, $\llbracket \mathbf{unit} \rrbracket$, $\llbracket \mathbf{int} \rrbracket$, $\llbracket \mathbf{real} \rrbracket$ are all their elements except \perp . The observable elements of $\llbracket \beta_1 \times \beta_2 \rrbracket$ are the pairs (b_1, b_2) where both b_1 and b_2 are observable. The observable elements of $\llbracket \beta_1 + \beta_2 \rrbracket$ are the elements $i_1 b$ or $i_2 b$, where b is observable.

For every observable value a of an observable type β , there is a generalized ISPCF term \underline{a} of type β such that $\llbracket \underline{a} \rrbracket \rho = a$ for every environment ρ . This extends our preexisting underlining notations.

Definition 10.4. The constants \underline{n} : **int** ($n \in \mathbb{Z}$), $\underline{*}$: **unit**, and \underline{a} ($a \in \mathbb{IR}$) are already defined. We let $\underline{(a, b)} \stackrel{\text{def}}{=} \langle \underline{a}, \underline{b} \rangle$, $\underline{(1, a)} \stackrel{\text{def}}{=} i_1 \underline{a}$, $\underline{(2, b)} \stackrel{\text{def}}{=} i_2 \underline{b}$.

In order to prevent any conflict of notations, we will require that the only preexisting constants \underline{f} of ISPCF of observable types are actually of basic types. For example, we do not allow any constant \underline{f} of type **int** \times **real** in our syntax.

Definition 10.5 (Strictly observable first-order constants). The set Σ consists of *strictly observable first-order constants* if and only if each constant \underline{f} in Σ :

- has a *first-order type*, namely one of the form $\sigma_1 \rightarrow \cdots \rightarrow \sigma_k \rightarrow \tau$, where each σ_i is a basic type and τ is an observable type; we call k the *arity* of \underline{f} , and we write it as $\alpha(\underline{f})$;
- its semantics f is *strictly observable*: $f(v_1) \cdots (v_k)$ is either observable or equal to \perp , and $f(v_1) \cdots (v_k) = \perp$ if v_1, \dots , or v_k is equal to \perp ;
- the only preexisting constants \underline{f} of ISPCF of observable types are of basic types, and their semantics f is observable.

All the constants we have used in our examples are first-order, for example $\underline{\log}$: **real** \rightarrow **real**, but also $\underline{\text{pos}}$: **real** \rightarrow **bool**, where **bool** = **unit** + **unit** is observable but not basic.

Our operational semantics will require two kinds of evaluation contexts. The point is that, given a generalized ISPCF term M of type $D\tau$, the computation starting from M will in general go through two phases $M \rightarrow^* \mathbf{ret} N \rightarrow^* \mathbf{ret} V$. The first phase $M \rightarrow^* \mathbf{ret} N$ will be allowed to use all rules, including rules that operate probabilistic choices, and

Probabilistic rules:

$$E^0[\mathbf{sample}[\mu]] \rightarrow E^0[\mathbf{ret} \underline{a}] \quad (a \in \mathbb{R}) \quad (5)$$

$$E^0[\mathbf{score} \underline{a}] \rightarrow E^0[\mathbf{ret} *] \quad (6)$$

$$E^0[\mathbf{do} x \leftarrow \mathbf{ret} M; N] \rightarrow E^0[N[x := M]]$$

Deterministic rules:

$$E[\mathbf{rec} M] \rightarrow E[M(\mathbf{rec} M)]$$

$$E[(\lambda x.M)N] \rightarrow E[M[x := N]]$$

$$E[\underline{f} \underline{a}_1 \cdots \underline{a}_k] \rightarrow E[\underline{f}(a_1) \cdots (a_k)] \quad (7)$$

if $k = \alpha(\underline{f}) \neq 0$ and $\underline{f}(a_1) \cdots (a_k)$ is observable

$$E[\pi_1 \langle M, N \rangle] \rightarrow E[M]$$

$$E[\pi_2 \langle M, N \rangle] \rightarrow E[N]$$

$$E[\mathbf{case}(\iota_1 M)NP] \rightarrow E[NM]$$

$$E[\mathbf{case}(\iota_2 M)NP] \rightarrow E[PM]$$

Fig. 3. The raw operational semantics

ends at some term of the form $\mathbf{ret} N$ (or loops forever); the second phase reduces N to a value V deterministically. The second phase applies (deterministic) rules under arbitrary evaluation contexts, while the first phase applies (arbitrary) rules under [certain contexts that we call](#) weak evaluation contexts.

Definition 10.6 (Evaluation contexts, values). The *evaluation contexts* E , the *weak evaluation contexts* E_0 , the *values* V and the *lazy values* V^0 are given by the following (pseudo-)grammar:

$$\begin{aligned} E &::= E^0 \mid \mathbf{ret} E \mid \langle E, M \rangle \mid \langle V, E \rangle \mid \iota_1 E \mid \iota_2 E \\ E^0 &::= [] \mid \mathbf{score} E^0 \mid E^0 N \mid \mathbf{do} x \leftarrow E^0; N \mid \underline{f} \underline{a}_1 \cdots \underline{a}_{i-1} E^0 \quad (i \leq \alpha(\underline{f})) \\ &\quad \mid \mathbf{case} E^0 NP \mid \pi_1 E^0 \mid \pi_2 E^0 \\ V &::= \underline{a} \mid \lambda x.M \mid \mathbf{ret} V \mid \langle V, V \rangle \mid \iota_1 V \mid \iota_2 V \mid \underline{f} \underline{a}_1 \cdots \underline{a}_i \quad (i < \alpha(\underline{f})) \\ V^0 &::= \underline{a} \mid \lambda x.M \mid \mathbf{ret} M \mid \langle M, N \rangle \mid \iota_1 M \mid \iota_2 M \mid \underline{f} \underline{a}_1 \cdots \underline{a}_i \quad (i < \alpha(\underline{f})) \end{aligned}$$

where \underline{a} ranges over the zero-ary constants of basic types, N, P range over generalized ISPCF terms, and $r \in \mathbb{R}$. The evaluation context $[]$ is the *hole*.

Evaluation contexts, weak evaluation contexts, values and lazy values are typed, although we will often omit mentioning it. The typing rules are as follows. Every value of the form \underline{a} with $\mathbf{a} \in \mathbb{IR}$ is of type **real**, and the type of the other values is their type as generalized ISPCF term. The types of evaluation contexts E (resp., E^0) are of the form $\sigma \vdash \tau$, in such a way that for every $M: \sigma$, $E[M]$ is of type τ . We omit the precise typing rules. The notation $E[M]$ stands for E where the hole $[]$ is replaced by M .

Using this, we define the *raw* operational semantics of ISPCF as the smallest binary relation \rightarrow between generalized ISPCF terms satisfying the clauses of Figure 3. The raw semantics does not mention any probabilistic information. This is dealt with by the following definition. We recall that $i: \mathbb{R} \rightarrow \mathbb{IR}_\perp$ is the embedding $r \mapsto [r, r]$.

Definition 10.7. Assume that all constants in Σ are first-order constants. The operational semantics of ISPCF is the map $Next: \Gamma \rightarrow \mathbf{VT}$ defined by:

- for every instance of a rule $L \rightarrow R$ of Figure 3 except (5), (6) and (7), $Next(L) = \delta_R$;
- (case of **sample**) $Next(E^0[\mathbf{sample}[\mu]])$ is the continuous valuation $f_{E^0}[\mu]$, where $f_{E^0}(a) \stackrel{\text{def}}{=} E^0[\mathbf{ret} \ a]$ for every $a \in \mathbb{R}$;
- (case of **score**) $Next(E^0[\mathbf{score} \ \underline{a}]) \stackrel{\text{def}}{=} |\underline{a}| \cdot \delta_{E^0[\mathbf{ret} \ *]}$;
- (case of f) Given $k = \alpha(f) \neq 0$, $Next(E[f \ \underline{a}_1 \cdots \underline{a}_k]) \stackrel{\text{def}}{=} \delta_{E[f(a_1) \cdots (a_k)]}$ if $f(a_1) \cdots (a_k)$ is observable, the constant zero valuation otherwise.

We let the reader check that every generalized ISPCF term parses in at most one way as the left-hand side of a rule of Figure 3, so that Definition 10.7 is non-ambiguous. To see this, one can write any generalized ISPCF term as $E[M_1]$, where E has maximal depth, in a unique way. If $M_1 = \mathbf{sample}[\mu]$, then only rule (5) applies, and only if E is weak. If $M_1 = \underline{a}$, then only rule (6) applies, and only if E is of the form $E^0[\mathbf{score}[]]$. If $M_1 = \mathbf{ret} \ M$ and $E = E^0[\mathbf{do} \ x \leftarrow []; N]$, then only the third rule applies. And so on, and we see that all the cases are mutually exclusive.

We relegate the proof of the next proposition, as well as that of other statements in this section, to Appendix B. That essentially consists in a pretty long verification of all cases.

PROPOSITION 10.8. *Let Σ consist of strictly observable first-order constants. The map $Next$ is Scott-continuous from Γ to \mathbf{VT} .*

As a corollary, $Next$ is Borel measurable from Γ to \mathbf{VT} , and is therefore a *kernel* in the sense of [81]; not just a measurable kernel, but a continuous kernel.

10.1 Soundness

Definition 10.9 (Redex, blocked term, normal form). A *redex* is a generalized ISPCF term that is of any of the ten forms found at the left of the rules of Figure 3. A *blocked term* is a generalized ISPCF term of the form $E[f \ \underline{a}_1 \cdots \underline{a}_k]$ where E is an evaluation context and $f(a_1) \cdots (a_k)$ is not observable. A *normal form* is a generalized ISPCF term that is neither a redex nor a blocked term.

All three categories are disjoint. For every blocked term or normal form L , $Next(L)$ is the zero valuation.

We will also need the following notions.

Definition 10.10 (Weak reduction, weak normal forms). A *weak instance* of a rule of Figure 3 is one where E , not just E^0 , is a weak evaluation context. A *weak redex* is a left-hand side of a weak instance of a rule. A *weak blocked term* is a generalized ISPCF term of the form $E^0[f \ \underline{a}_1 \cdots \underline{a}_k]$ where E^0 is a weak evaluation context and $f(a_1) \cdots (a_k)$ is not observable. A *weak normal form* is a generalized ISPCF term that is neither a weak redex nor a weak blocked term.

LEMMA 10.11. *Let Σ consist of strictly observable first-order constants. The weak normal forms are exactly the lazy values.*

Let $Norm_\tau$ be the set of normal forms of type τ .

LEMMA 10.12. *Let Σ consist of strictly observable first-order constants. For every type τ , $Norm_\tau$ is exactly the set of values of type τ .*

The following lemma will apply to $\kappa \stackrel{\text{def}}{=} \text{Next}$, where $X \stackrel{\text{def}}{=} \Gamma$, and where Norm is the set of normal forms. For every $x \in X$, for every open subset U of Norm , $\text{Next}^{\leq n}(x)(U)$ is the ‘probability’ that we will reach U , starting from x , in at most n Next steps, and $\text{Next}^*(x)(U)$ is the ‘probability’ that we will reach U starting from x in any number of steps. (The quotes around ‘probability’ reflect the fact that those probabilities need not be bounded by 1.) This way of defining ‘probabilities’ of reaching U is inspired from Lemma 3.9 and Equation (6) of [17]. Let us also recall the restriction $\mu|_U$ of a continuous valuation to an open subset U , defined by $\mu|_U(V) \stackrel{\text{def}}{=} \mu(U \cap V)$.

LEMMA 10.13. *Let X be a topological space, and let Norm and $\overline{\text{Norm}}$ be two complementary open subsets of X . Let $\kappa: X \rightarrow \mathbb{V}X$ be a continuous map, and let us assume that for every $x \in \text{Norm}$, $\kappa(x)$ is the zero valuation. We define:*

- $\kappa^{\leq 0}$ as mapping every $x \in \text{Norm}$ to δ_x , and every $x \in \overline{\text{Norm}}$ to the zero valuation;
- $\kappa^{\leq 1} \stackrel{\text{def}}{=} \kappa + \kappa^{\leq 0}$;
- $\kappa^{\leq n+1} \stackrel{\text{def}}{=} (\kappa^{\leq n})^\dagger \circ \kappa^{\leq 1}$, for every $n \geq 1$.

Then:

- (1) the maps $\kappa^{\leq n}$ are continuous from X to $\mathbb{V}X$;
- (2) for every $x \in X$, the family $(\kappa^{\leq n}(x)|_{\text{Norm}})_{n \geq 1}$ is monotonically increasing;
- (3) the formula $\kappa^*(x) \stackrel{\text{def}}{=} \sup_{n \geq 1} \kappa^{\leq n}(x)|_{\text{Norm}}$ defines a continuous map κ^* from X to $\mathbb{V}X$.

From now on, we let Norm be the subset of configurations (M, θ) such that $M\theta$ is a normal form. Being a normal form is a property of shapes, namely any two generalized ISPCF terms with the same shape will both be normal, or neither of them will be. It follows that Norm and its complement $\overline{\text{Norm}}$ are open in Γ . Let us give another characterization of $\text{Next}^{\leq n}$.

LEMMA 10.14. *Let Σ consist of strictly observable first-order constants. For every generalized ISPCF term M , for every $n \in \mathbb{N}$, we have:*

- (1) if $M \in \text{Norm}$, then $\text{Next}^{\leq n}(M) = \delta_M$;
- (2) if $M = E^0[\text{sample}[\mu]]$, and $n \geq 1$, then for every $U \in \mathcal{O}\Gamma$, $\text{Next}^{\leq n}(M)(U) = \int_{a \in \mathbb{R}} \text{Next}^{\leq n-1}(E^0[\text{ret } \underline{a}])(U) d\mu$;
- (3) if $M = E^0[\text{score } \underline{a}]$ and $n \geq 1$, then $\text{Next}^{\leq n}(M) = |\underline{a}| \cdot \text{Next}^{\leq n-1}(E^0[\text{ret } *])$;
- (4) if $M = E[f \ \underline{a}_1 \cdots \underline{a}_k]$ and $n \geq 1$, then $\text{Next}^{\leq n}(M)$ is equal to $\text{Next}^{\leq n-1}(E[f(a_1) \cdots (a_k)])$ if $f(a_1) \cdots (a_k)$ is observable, to the zero valuation otherwise;
- (5) for every instance of a rule $L \rightarrow R$ of Figure 3 except (5), (6) and (7), if $n \geq 1$ then $\text{Next}^{\leq n}(L) = \text{Next}^{\leq n-1}(R)$.

We extend the denotational semantics of Figure 2 to generalized ISPCF terms by positing:

$$\llbracket \underline{a} \rrbracket \rho \stackrel{\text{def}}{=} \mathbf{a},$$

for every $\mathbf{a} \in \mathbb{I}\mathbb{R}$. We note that $\llbracket M \rrbracket \rho$ is independent of ρ . In the sequel, we will therefore simply write $\llbracket M \rrbracket$ for the semantics of the generalized ISPCF term M , disregarding the useless ρ . Alternatively, every generalized ISPCF term M can be written as $P\theta$ for some unique configuration (P, θ) , and $\llbracket M \rrbracket$ is also equal to $\llbracket P \rrbracket \theta$, reading θ as an environment. The Scott-continuity of $\llbracket P \rrbracket$ immediately entails the following.

LEMMA 10.15. *Let Γ_τ be the subspace of Γ consisting of generalized ISPCF terms of type τ . The map $\llbracket _ \rrbracket : \Gamma_\tau \rightarrow \llbracket \tau \rrbracket$ is Scott-continuous.*

It follows that, given any Scott-open subset U of $\llbracket \tau \rrbracket$, for any type τ , the set \overline{U} of normal forms M of type τ such that $\llbracket M \rrbracket \in U$ is Scott-open in Γ_τ . We make that into a definition, and we also introduce another open set $\overline{\overline{U}}$.

Definition 10.16 (The open sets $\overline{U}, \overline{\overline{U}}$). For every type τ , for every Scott-open subset U of $\llbracket \tau \rrbracket$, let \overline{U} be the Scott-open set of normal forms V of type τ such that $\llbracket V \rrbracket \in U$, and $\overline{\overline{U}}$ be the Scott-open set of terms of the form $\mathbf{ret} V$ with $V \in \overline{U}$.

It is clear that $\overline{\overline{U}}$ is Scott-open: by definition of the topology of Γ , the map $M \mapsto \mathbf{ret} M$ is a homeomorphism of Γ onto the subspace of those configurations starting with \mathbf{ret} .

PROPOSITION 10.17 (SOUNDNESS). *Let Σ consist of strictly observable first-order constants. For every generalized ISPCF term $M: D\tau$, where τ is any type, for every open subset U of $\llbracket \tau \rrbracket$, $\llbracket M \rrbracket(U) \geq \text{Next}^*(M)(\overline{\overline{U}})$.*

10.2 Adequacy

Adequacy means that the inequality of Proposition 10.17 can be reinforced to an equality, provided that τ is an observable type.

We recall that the *deterministic rules* are all those of Figure 3 except the first three, and that the *weak rules* are those where the evaluation context is weak. We say that $M \rightarrow^* N$ by *deterministic rules* (resp., by *weak deterministic rules*) if and only if one can connect M to N by a finite string of instances of deterministic rules (resp., weak deterministic rules).

In order to establish adequacy, we will use a logical relation $(R_\tau)_\tau$ type in the same style as those used in [32, 33]. Before we do this, we need the following notion.

Definition 10.18 (Observable open set). An *observable open subset* of $\llbracket \beta \rrbracket$ is a Scott-open subset of $\llbracket \beta \rrbracket$ consisting of observable elements.

For each type τ , R_τ will be a binary relation between generalized ISPCF terms of type τ and elements of $\llbracket \tau \rrbracket$. We define it with the help of auxiliary relations $R_{\tau \vdash D\beta}^\perp$. The latter relates evaluation contexts of type $D\tau \vdash D\beta$ and Scott-continuous maps from $\llbracket \tau \rrbracket$ to $\llbracket D\beta \rrbracket$, where β is an observable type. We will use short phrases such as “for all $E^0 R_{\tau \vdash D\beta}^\perp h$ ” instead of “for every observable type β , for every weak evaluation context E^0 of type $D\tau \vdash D\beta$, for every Scott-continuous map h from $\llbracket \tau \rrbracket$ to $\llbracket D\beta \rrbracket$, if $E^0 R_{\tau \vdash D\beta}^\perp h$ then”.

Definition 10.19 (Logical relation, $R_\tau, R_{\tau \vdash D\beta}^\perp$). We define:

- for every *basic* type τ , $M R_\tau a$ if and only if either $a = \perp$, or $M \rightarrow^* \underline{b}$ by weak deterministic rules, for some zero-ary constant \underline{b} such that $b \geq a$.
- $M R_{\tau_1 \times \tau_2} a$ if and only if $a = \perp$, or $a = (a_1, a_2)$ where $M \rightarrow^* \langle M_1, M_2 \rangle$ by weak deterministic rules, for some $M_1 R_{\tau_1} a_1$ and $M_2 R_{\tau_2} a_2$.
- $M R_{\tau_1 + \tau_2} a$ if and only if $a = \perp$, or $M \rightarrow^* \iota_i N$ by weak deterministic rules and $a = (i, b)$, for some $i \in \{1, 2\}$ and some $N R_{\tau_i} b$ with $b \neq \perp$.
- $M R_{\sigma \rightarrow \tau} g$ if and only if for all $N R_\sigma a$, $MN R_\tau g(a)$.
- $M R_{D\tau} \mu$ if and only if for all $E^0 R_{\tau \vdash D\beta}^\perp h$, for every observable open subset U of $\llbracket \beta \rrbracket$, $\text{Next}^*(E^0[M])(\overline{\overline{U}}) \geq h^\dagger(\mu)(U)$;
- $E^0 R_{\tau \vdash D\beta}^\perp h$ if and only if for all $M R_\tau a$, for every observable open subset U of $\llbracket \beta \rrbracket$, $\text{Next}^*(E^0[\mathbf{ret} M])(\overline{\overline{U}}) \geq h(a)(U)$.

This is a well-founded definition by induction on the size of τ . Note that $R_{D\tau}$ is defined in terms of $R_{\tau \vdash D\beta}^\perp$, which itself depends on R_τ ; there is no dependency on, say, $R_{D\beta}$, which would make the definition ill-founded.

We let $M R_\tau$ abbreviate the set of all values $a \in \llbracket \tau \rrbracket$ such that $M R_\tau a$. One can show that, if Σ consists of strictly observable first-order constants, the following properties hold, for all relevant M, M', τ (see Appendix B.3 for proofs): (i) if $M \rightarrow^* M'$ by weak deterministic rules, then $M R_\tau a$ if and only if $M' R_\tau a$ (Lemma B.5); (ii) $M R_\tau$ is a non-empty Scott-closed subset of $\llbracket \tau \rrbracket$ (Lemma B.6); (iii) $M R_\tau$ is downwards-closed; (iv) \perp is in $M R_\tau$ (easy consequences of (ii)); (v) for every $a \in \llbracket \beta \rrbracket$ where β is an observable type, $\underline{a} R_\beta a$ (Lemma B.7).

The fundamental lemma of logical relations states that terms are related to their semantics. We prove the variant suited to our case below. Given an ISPCF term M , a *substitution* θ for M is a map from the free variables of M to generalized ISPCF terms of the same type, and $M\theta$ denotes the result of application θ to M . Given a substitution θ and an environment ρ with the same domain D , we write $\theta R_* \rho$ to mean that $\theta(x_\sigma) R_\sigma \rho(x_\sigma)$ for every $x_\sigma \in D$. The proof is an induction on the size of M . It relies on (ii) above for terms of the form **rec** N , and requires the weaker forms (iii) and (iv) for first-order constants, projections, and **case** statements; (v) is needed in the case of first-order constants and terms of the form **sample** $[\mu]$, and (i) is used in each and every case of the induction.

PROPOSITION 10.20. *Let Σ consist of strictly observable first-order constants. For every ISPCF term $M: \tau$, for all $\theta R_* \rho$, $M\theta R_\tau \llbracket M \rrbracket \rho$.*

We obtain the generalized ISPCF terms as $M\theta$, where (M, θ) ranges over Γ . In that case, for every variable x in the domain of θ , x has type **real**, and if we write \underline{a} for $\theta(x)$, then we may define $\rho(x)$ as \mathbf{a} . Then $\llbracket M \rrbracket \rho$ is the semantics $\llbracket M\theta \rrbracket$ of the generalized ISPCF term $M\theta$. By (v), $\underline{a} R_{\mathbf{real}} \mathbf{a}$, so $\theta R_* \rho$. Proposition 10.20 then implies the following.

COROLLARY 10.21. *Let Σ consist of strictly observable first-order constants. For every generalized ISPCF term $M: \tau$, $M R_\tau \llbracket M \rrbracket$.*

LEMMA 10.22. *Let Σ consist of strictly observable first-order constants. For every observable type β , for every generalized ISPCF term $N: \beta$, for every $b \in \llbracket \beta \rrbracket$, if $N R_\beta b$ then either b is not observable, or b is observable and $N \rightarrow^* V$ by deterministic rules for some value $V: \beta$ such that $\llbracket V \rrbracket \geq b$.*

Note that $N \rightarrow^* V$ by deterministic rules, but not necessarily weak rules.

We obtain the following promised adequacy theorem by considering the empty weak evaluation context $E^0 \stackrel{\text{def}}{=} []$, and showing that $E^0 R_{\beta + D\beta}^\perp \eta_{\llbracket \beta \rrbracket}$, using Lemma 10.22 and Lemma 10.14, item 1; we conclude by Corollary 10.21.

THEOREM 10.23 (ADEQUACY). *Let Σ consist of strictly observable first-order constants. For every observable type β , for every generalized ISPCF term $M: D\beta$, for every observable open subset U of $\llbracket \beta \rrbracket$, $\llbracket M \rrbracket (U) = \text{Next}^*(M)(\overline{\overline{U}})$.*

Remark 10.2. Why do we restrict M to be of type $D\beta$ in Theorem 10.23? Restricting it to be of observable type β would be silly, because only deterministic computations happen at type β . In order to be able to do any probabilistic computation at all, M has to be of distribution type. A similar choice was made in [32, 33].

Remark 10.3. The reason why Theorem 10.23 only considers *observable* open subsets U is that adequacy is concerned with *terminating* computations, and terms of observable type β terminate on values V (if they do terminate at all), whose semantics is observable. For non-observable open subsets U , the denotational semantics gives us more information. For example, when U is the whole of $\llbracket \beta \rrbracket$, $\text{Next}^*(M)(\overline{\overline{U}})$ is the ‘probability’ that M terminates at all, while $\llbracket M \rrbracket (U)$ is larger in general, and represents the ‘probability’ of all executions starting from M , including those that do not terminate. For example, let $M \stackrel{\text{def}}{=} \mathbf{ret}(\mathbf{rec}(\lambda n_{\mathbf{int}}.n))$, then $\text{Next}^*(M)(\overline{\overline{\llbracket \mathbf{int} \rrbracket}}) = 0$, while $\llbracket M \rrbracket (\llbracket \mathbf{int} \rrbracket) = \delta_\perp(\llbracket \mathbf{int} \rrbracket) = 1$. For a product type $\beta_1 \times \beta_2$, the Scott-open set $U \stackrel{\text{def}}{=} \llbracket \beta_1 \rrbracket \times U'$, where U' is a non-empty observable subset of $\llbracket \beta_2 \rrbracket$, is

not observable either, but is smaller than $\llbracket \beta_1 \times \beta_2 \rrbracket$. In that case, $\text{Next}^*(M)(\overline{U})$ is the ‘probability’ that M terminates on a value $\mathbf{ret}\langle V_1, V_2 \rangle$ with $V_2 \in \overline{U}'$, but $\llbracket M \rrbracket(U)$ is the ‘probability’ that $\pi_2 M$ terminates with a value in U' , thereby also counting all computations for which $\pi_1 M$ does not terminate. Another view to the question is given by Lemma 10.24 below.

Remark 10.4. The reader interested in extending the adequacy theorem to non-observable open subsets U should be warned of the following difficulty. Let U be the whole of $\llbracket \beta \rrbracket$. While $\llbracket M \rrbracket(U)$ is perfectly well-defined, what would be the operational meaning of “the ‘probability’ of all executions starting from M ”? This is difficult, already for executions not involving $\mathbf{sample}[\mu]$ and rule (5), but involving \mathbf{score} and rule (6). Imagine an infinite execution starting for M and calling \mathbf{score} on real values a_1, a_2, \dots , in succession. One would imagine that the ‘probability’ of the unique execution starting from M is $\prod_{i=1}^{+\infty} a_i$. However, that is ill-defined in general. For an example, consider $a_i \stackrel{\text{def}}{=} 2$ for i odd, $1/2$ for i even.

We say that Σ is *terse* if and only if for every $n \in \mathbb{Z}$, there is a unique zero-ary constant \underline{n} of type \mathbf{int} such that $\llbracket \underline{n} \rrbracket = n$, and $\underline{*}$ is the only zero-ary constant of type \mathbf{unit} . In this case, we can avoid the use of sets of the form \overline{U} in the definition of \lesssim^{app} and \cong^{app} , because of the following lemma. Note that we do not need to require a similar condition of terseness on \mathbf{void} (if Σ consists of strictly observable first-order constants, then there can be no zero-ary constant of type \mathbf{void}), or on \mathbf{real} (because of the way we defined generalized ISPCF terms at the beginning of Section 10).

LEMMA 10.24. *Let Σ be terse and consist of strictly observable first-order constants. For every observable type β , the map $M \mapsto \llbracket M \rrbracket$ is an order isomorphism from $Norm_\beta$ to the subdcpo Obs_β of observable elements in $\llbracket \beta \rrbracket$. The open subsets of $Norm_\beta$ are exactly the sets of the form \overline{U} , where U is a observable Scott-open subset of $\llbracket \beta \rrbracket$. Those of $Norm_{D\beta}$ are exactly the sets of the form $\overline{\overline{U}}$, where U is a observable Scott-open subset of $\llbracket \beta \rrbracket$.*

PROOF. The first claim is clear. The inverse map is the map $a \mapsto \underline{a}$ of Definition 10.4. For the second claim, Obs_β is a Scott-open subset of $\llbracket \beta \rrbracket$, and is in particular a subdcpo (i.e., directed suprema are computed as in $\llbracket \beta \rrbracket$). Let U' be any open subset of $Norm_\beta$. Its image by the map $a \mapsto \underline{a}$ is an open subset of $\llbracket \beta \rrbracket$ included in Obs_β , hence is Scott-open in Obs_β , since Obs_β is a subdcpo of $\llbracket \beta \rrbracket$. The third claim follows from the fact that $M \mapsto \mathbf{ret} M$ is an isomorphism between $Norm_\beta$ and $Norm_{D\beta}$. \square

10.3 Contextual equivalence and the applicative preorder

An important, and expected, consequence of adequacy, is that equality of denotations implies contextual equivalence, namely that for any two generalized ISPCF terms M, N of any type τ , if $\llbracket M \rrbracket = \llbracket N \rrbracket$ then M and N are contextually equivalent, meaning that every observable type β , for every context \mathcal{E} of type $\tau \vdash D\beta$, the ‘probability’ that $\mathcal{E}[M]$ evaluates to a value in any given open set U of normal forms is equal to the ‘probability’ that $\mathcal{E}[N]$ evaluates to a value in U .

We define the required contexts \mathcal{E} as generalized ISPCF terms with arbitrarily many holes. This represents the fact that one may replace M at any number of positions in a program by a contextually equivalent program N and not change the observable behavior of the program.

In order to formalize this, there is no need to invent a new notion of multi-hole context. Indeed, replacing all the holes in \mathcal{E} with a term M gives a term with the same semantics as the application QM , where $Q \stackrel{\text{def}}{=} \lambda x. \mathcal{E}[x]$. Hence the notion of contextual equivalence we are looking is the following one, usually called *applicative equivalence*. This is the equivalence relation associated with a more primitive notion of *applicative preorder*, which we now define.

Definition 10.25 (Applicative preorder \lesssim^{app} and equivalence). For any two generalized ISPCF terms M, N of the same type τ , $M \lesssim^{app} N$ (resp., $M \cong^{app} N$) if and only if, for every observable type β , for every generalized ISPCF term $Q: \tau \rightarrow D\beta$, for every open subset U of $Norm_{D\beta}$, $Next^*(QM)(U) \leq Next^*(QN)(U)$ (resp., $=$).

The adequacy Theorem 10.23 allows us to simplify the definition of \lesssim^{app} and \cong^{app} as follows. Note that, if $Q = \lambda x_\tau. \mathcal{E}[x]$, where \mathcal{E} is one of our purported multi-hole contexts, then $\llbracket QM \rrbracket = \llbracket \mathcal{E}[M] \rrbracket$, justifying the view that the applicative preorder is really about comparing programs with zero, one, or several occurrences of M replaced by another term N .

PROPOSITION 10.26. *Let Σ be terse and consist of strictly observable first-order constants. For all generalized ISPCF terms M, N of the same type τ , $M \lesssim^{app} N$ (resp., $M \cong^{app} N$) if and only if, for every observable type β , for every generalized ISPCF term $Q: \tau \rightarrow D\beta$, for every observable Scott-open subset U of $\llbracket \beta \rrbracket$, $\llbracket QM \rrbracket(U) \leq \llbracket QN \rrbracket(U)$.*

PROOF. By Lemma 10.24, one may replace the quantification over open subsets U of $Norm_{D\beta}$ by a quantification over open subsets of the form $\overline{\overline{U}}$, where U ranges over the observable Scott-open subsets. Then, by Theorem 10.23, $Next^*(QM)(\overline{\overline{U}}) = \llbracket QM \rrbracket(U)$, and similarly with QN . \square

Importantly, we obtain the following easy result.

PROPOSITION 10.27. *Let Σ be terse and consist of strictly observable first-order constants. Let M and N be two generalized ISPCF terms of the same type τ . The following implications hold:*

- (1) $\llbracket M \rrbracket \leq \llbracket N \rrbracket \Rightarrow M \lesssim^{app} N$;
- (2) $\llbracket M \rrbracket = \llbracket N \rrbracket \Rightarrow M \cong^{app} N$.

PROOF. If $\llbracket M \rrbracket \leq \llbracket N \rrbracket$, then for every observable type β , for every generalized ISPCF term $Q: \tau \rightarrow D\beta$, $\llbracket QM \rrbracket = \llbracket Q \rrbracket(\llbracket M \rrbracket) \leq \llbracket Q \rrbracket(\llbracket N \rrbracket) = \llbracket QN \rrbracket$. Using Proposition 10.26, $M \lesssim^{app} N$. The second part of the theorem is an easy consequence of the first part. \square

The converse implications would be a form of full abstraction, and are quite probably hopeless. Therefore we will not bother with them.

Remark 10.5. Following up on Remark 8.1, it follows that when x is not free in N and y is not free in M , the terms $\mathbf{do} x \leftarrow M; \mathbf{do} y \leftarrow N; P$ and $\mathbf{do} y \leftarrow N; \mathbf{do} x \leftarrow M; P$ are contextually equivalent.

Remark 10.6. Similarly, the terms `normal`, `do <x, y> ← box_muller; ret x`, and `do <x, y> ← box_muller; ret y` (see Section 9.2) are contextually equivalent. The terms `expo`, `expo'` and `von_neumann` of Section 9.4 and Appendix A.3 are contextually equivalent, too.

11 A PRECISE OPERATIONAL SEMANTICS

We give a second operational semantics that is closer to the intent of [10, 17, 81]. This requires slightly stricter assumptions than simply having strictly observable first-order constants. Those assumptions are always met in practice, as far as we know. Note that `exp`, `sin`, `cos`, but also `pos` and `log`, are precise, for example. One instance of a non-precise constant would be the sign map, if we decide to define it as \hat{g} (see Section 8), where g maps every negative number to 0 and every non-negative number to 1; indeed, \hat{g} maps every interval $[a, b]$ with $a \leq 0 \leq b$ to $[0, 1]$. We can define the (precise) sign map as $\lambda x_{\mathbf{real}}. \mathbf{if}(\underline{\text{pos}} x) \underline{1.0} \underline{0.0}$ instead.

Definition 11.1. The set Σ consists of *precise first-order constants* if and only if Σ has strictly observable first-order constants, and additionally, for each constant $\underline{f}: \sigma_1 \rightarrow \cdots \rightarrow \sigma_k \rightarrow \tau$ of arity k in Σ , f is *precise*, namely:

- for all maximal elements $v_1 \in \llbracket \sigma_1 \rrbracket, \dots, v_k \in \llbracket \sigma_k \rrbracket$, $f(v_1) \cdots (v_k)$ is either maximal in $\llbracket \tau \rrbracket$ or equal to \perp .

When $\sigma_1 = \cdots = \sigma_k = \tau = \mathbf{real}$, this means that f encodes a partial continuous map from \mathbb{R}^k to \mathbb{R} . If we equate $i(v) = [v, v]$ with the real number v , the domain $\{(v_1, \dots, v_k) \in \mathbb{R}^k \mid f(v_1) \cdots (v_k) \neq \perp\}$ of f is an open subset of \mathbb{R}^k , and f is required to be continuous on that domain.

A *precise configuration* (M, θ) is a configuration where for every $x \in \text{fv}(M)$, $\theta(x)$ is a maximal element of \mathbb{IR}_\perp , namely a real number.

A *precise generalized ISPCF term* is a term $M\theta$ obtained from a precise configuration (M, θ) . Given a shape M , one can equate θ with an element of $\mathbb{R}^{\text{fv}(M)}$, namely with a finite tuple of real numbers. $\mathbb{R}^{\text{fv}(M)}$, and by extension the set of precise configurations with shape M , is equipped with its usual metric topology, which we will call its *Euclidean topology*.

Definition 11.2 (Γ^0). We topologize Γ^0 as the set of all precise configurations, topologized as the coproduct over all shapes M of the spaces of precise configurations with shape M , each being given the Euclidean topology.

On the one hand, Γ^0 is a much more familiar space to the ordinary topologist. If we assume a countable set Σ of constants and a countable set \mathcal{M} of measures, then there are countably many configurations. Then Γ^0 is a countable coproduct of Polish spaces, hence it is itself Polish. (We will not make use of this fact.) On the other hand, Γ^0 is also a topological subspace of Γ . We write ι for the subspace embedding; we have $\iota(M, \theta) = (M, i \circ \theta)$, where i is the familiar subspace embedding $r \mapsto [r, r]$ of \mathbb{R} into \mathbb{IR}_\perp .

LEMMA 11.3. *Assume that Σ consists of precise first-order constants. There is a unique continuous map $\text{Next}^0: \Gamma^0 \rightarrow \mathbf{V}\Gamma^0$ such that, for every $(M, \theta) \in \Gamma^0$, $\iota[\text{Next}^0(M, \theta)] = \text{Next}(\iota(M, \theta))$.*

PROOF. By inspection of Definition 10.7. The key point is the case of \underline{f} , where the result holds because Σ has precise first-order constants. \square

Remark 11.1. In general, $\text{Next}(M\theta)$ is a minimal valuation on Γ . However, $\text{Next}^0(M, \theta)$ need not be a minimal valuation. For example, $\text{Next}^0(x, [x := \mathbf{sample}[\lambda]])$ is, up to isomorphism, the Lebesgue valuation on \mathbb{R} , which is not minimal by Lemma 6.1.

Next^0 defines a kernel. Using Lemma 10.13, we obtain a continuous map $\text{Next}^{0,*}: \Gamma^0 \rightarrow \mathbf{V}\Gamma^0$, provided we replace Norm by the set $\text{Norm}^0 \stackrel{\text{def}}{=} \text{Norm} \cap \Gamma^0$. By induction on n , it is easy to see that $\iota[\text{Next}^{0,* \leq n}(M, \theta)] = \text{Next}^{\leq n}(\iota(M, \theta))$. It follows:

LEMMA 11.4. *Assume that Σ consists of precise first-order constants. For every precise configuration (M, θ) , $\iota[\text{Next}^{0,*}(M, \theta)] = \text{Next}^*(\iota(M, \theta))$.*

Proposition 10.17 and Theorem 10.23 then immediately imply the following. We equate Γ_0 with a subspace of Γ , and omit any mention of the embedding ι .

THEOREM 11.5. *Let Σ consist of precise first-order constants. For every precise generalized ISPCF term $M: D\tau$, where τ is any type, for every open subset U of $\llbracket \tau \rrbracket$, $\llbracket M \rrbracket(U) \geq \text{Next}^{0,*}(M)(\overline{U} \cap \Gamma^0)$. This inequality is an equality if τ is an observable type and U is observable.*

In particular, let us consider the observable type **real**, and let us examine what Theorem 11.5 entails about $\llbracket M \rrbracket$ for $M: \mathbf{Dreal}$. We first enrich the language, if necessary, so as to include constants \underline{a} for every real number a . This is harmless: this will not change the value of $\llbracket M \rrbracket$, and this preserves the property of having precise first-order constants. This certainly makes the language uncountable, but, although this would seem peculiar from a programmer's point of view, this is perfectly allowed. (The only point in this paper where we have mentioned some countability restriction on the language is when we commented about the possible Polishness of Γ^0 after Definition 11.2.)

The observable open subsets U of $\llbracket \mathbf{real} \rrbracket = \mathbb{IR}_\perp$ are all the proper open subsets of \mathbb{IR}_\perp , namely all of them except \mathbb{IR}_\perp itself, or equivalently, the Scott-open subsets U of \mathbb{IR}_\perp that do not contain \perp . $\overline{\overline{U}} \cap \text{Config}^0$ is the Scott-open set of precise configurations (**ret** $x_1, [x_1 := a]$), where \underline{a} ranges over the constants of type **real** such that $a \in U$. (We equate a with $i(a) = [a, a]$.) Then Theorem 11.5 states that for every precise generalized ISPCF term $M: \mathbf{real}$, $\llbracket M \rrbracket(U) = \text{Next}^{0,*}(M)(\overline{\overline{U}} \cap \Gamma^0)$. Now, if U and V are two proper open subsets of \mathbb{IR}_\perp that agree on \mathbb{R} , namely such that $i^{-1}(U) = i^{-1}(V)$, then $\overline{\overline{U}} \cap \Gamma^0 = \overline{\overline{V}} \cap \Gamma^0$, so $\llbracket M \rrbracket(U) = \llbracket M \rrbracket(V)$.

Let U_0 be the Scott-open set $\mathbb{IR}_\perp \setminus \{\perp\}$, and ν be the restriction $\llbracket M \rrbracket|_{U_0}$: $\nu(U) \stackrel{\text{def}}{=} \llbracket M \rrbracket(U \cap U_0)$ for every Scott-open subset U of \mathbb{IR}_\perp . Then $\llbracket M \rrbracket = \nu + r.\delta_\perp$, where $r \stackrel{\text{def}}{=} \llbracket M \rrbracket(\mathbb{IR}_\perp) - \llbracket M \rrbracket(U_0)$, or $r \stackrel{\text{def}}{=} +\infty$ if $\llbracket M \rrbracket(\mathbb{IR}_\perp) = +\infty$. The fact that $\llbracket M \rrbracket(U) = \llbracket M \rrbracket(V)$ for all proper open subsets of \mathbb{IR}_\perp that agree on \mathbb{R} entails that $\nu(U) = \nu(V)$ for all open subsets of \mathbb{IR}_\perp (not necessarily proper) that agree on \mathbb{R} . By Proposition 5.2 of [34], there is a unique continuous valuation μ on \mathbb{R} such that $\nu = i[\mu]$, and therefore $\llbracket M \rrbracket = i[\mu] + r.\delta_\perp$. We have therefore established the following.

COROLLARY 11.6. *Let Σ consist of precise first-order constants. For every precise generalized ISPCF term $M: \mathbf{Dreal}$, $\llbracket M \rrbracket$ is of the form $i[\mu] + r.\delta_\perp$ for some continuous valuation μ on \mathbb{R} and some $r \in \overline{\mathbb{R}}_+$.*

In other words, under these assumptions, the semantics of $M: \mathbf{Dreal}$ really is a **distribution** μ on the space \mathbb{R} of real numbers (not just intervals), plus some ‘probability’ r of non-terminating.

12 A SAMPLING-BASED OPERATIONAL SEMANTICS

An operational semantics is usually an abstract view of an implementation. However, operational semantics such as those of previous sections (and such as those of [10, 17, 81], which even go as far as manipulating *true* real numbers) are far from an implementation. Park, Pfenning and Thrun [62], and later Dal Lago and Hoshino [10] introduce a more concrete *sampling-based* operational semantics. Roughly speaking, it is more concrete in the sense that it no longer has to sample from arbitrary measures on \mathbb{R} , and instead draws arbitrarily long strings of independent, uniform random bits, which are drawn at random prior to execution.

We use a similar idea here. This will apply under some reasonable assumptions, the most important one being that we only have one term of the form **sample** $[\mu]$, namely **sample** $[\lambda_1]$. As we have hinted in Section 9, one can define quite a number of other distributions from that one alone.

12.1 Random strings

Our sampling-based operational semantics will be a *deterministic* semantics, parameterized by an infinite string of bits, which we can think as having been drawn at random, uniformly, in advance. This trick is routinely used in complexity theory, for example, where a popular definition of a randomized Turing machine is a Turing machine with an additional read-only *random tape* on which the head can only move right. It is also at the heart of the definition of computable probability theory, where a *computable random variable* on a computable metric space S is a measurable

map $X: \{0, 1\}^{\mathbb{N}} \rightarrow S$ such that X is computable on a subset of $\{0, 1\}^{\mathbb{N}}$ of ν -measure one [65, Definition II.12]. We have also used similar tricks in Section A.5, too.

We will reuse several notions and notations from Section 9.5. Notably, ν is the uniform measure $\text{bin}[\lambda_1]$ on $\{0, 1\}^{\mathbb{N}}$. We also recall the maps $\text{bin}: \mathbb{R} \rightarrow \{0, 1\}^{\mathbb{N}}$, $\text{num}: \{0, 1\}^{\mathbb{N}} \rightarrow \mathbb{R}$, $\text{split}: \{0, 1\}^{\mathbb{N}} \rightarrow (\{0, 1\}^{\mathbb{N}})^2$, the notation $s[m] \stackrel{\text{def}}{=} (s_{\langle m, n \rangle})_{n \in \mathbb{N}}$ and the shift $\uparrow s$.

12.2 The sampling-based operational semantics

Our next semantics will be a partial map $\text{Next}'(s): \Gamma' \rightarrow \Gamma'$, parameterized by a fixed random string s , and where Γ' is a space of **objects that we call enriched configurations**. This is somewhat similar to the sampling-based operational semantics of [10, Section 3.3]. An enriched configuration is a tuple (M, i, r) where M is a generalized ISPCF term, $i \in \mathbb{N}$, and $r \in \mathbb{R}_+$: i is the number of the next random string $s[i]$ to use in order to implement rule (5), and r is the product of the score values produced by rule (6) so far. We let Γ' be the space of all indexed configurations. This is a product $\text{dcpo } \Gamma \times \mathbb{N} \times \mathbb{R}_+$, where \mathbb{N} is ordered by equality and \mathbb{R}_+ by its usual ordering. The Scott topology on the product coincides with the product topology, since this is a finite product of continuous posets [31, Proposition 5.1.54]; as such, Γ' is also a continuous poset. Random strings are elements of the topological product $\{0, 1\}^{\mathbb{N}}$, where $\{0, 1\}$ has the discrete topology.

Definition 12.1 (Sampling-based operational semantics). Assume that all constants in Σ are first-order constants, and that the only term of the form **sample** $[\mu]$ is **sample** $[\lambda_1]$, where λ_1 is Lebesgue measure on $[0, 1]$. The *sampling-based operational semantics* of ISPCF is the partial map $\text{Next}': \{0, 1\}^{\mathbb{N}} \rightarrow \Gamma' \rightarrow \Gamma'$ defined by:

- for every instance of a rule $L \rightarrow R$ of Figure 3 except (5), (6) and (7), $\text{Next}'(s)(L, i, r) = (R, i, r)$;
- (case of **sample**) $\text{Next}'(s)(E[\mathbf{sample}[\lambda_1]], i, r) \stackrel{\text{def}}{=} (E[\mathbf{ret num}(s[0])], i + 1, r)$;
- (case of **score**) $\text{Next}'(s)(E[\mathbf{score } \underline{a}], i, r) \stackrel{\text{def}}{=} (E[\mathbf{ret } *], i, |\underline{a}| \cdot r)$;
- (case of \underline{f}) If $k = \alpha(\underline{f}) \neq 0$, then $\text{Next}'(s)(E[\underline{f } \underline{a}_1 \cdots \underline{a}_k], i, r) \stackrel{\text{def}}{=} (E[\underline{f}(a_1) \cdots (a_k)], i, r)$ if $\underline{f}(a_1) \cdots (a_k)$ is observable;
- $\text{Next}'(s)(M, i, r)$ is undefined otherwise.

Let us write $(M, i, r) \rightarrow_s (M', i', r')$ for “ $\text{Next}'(s)(M, i, r)$ is defined and equal to (M', i', r') ”. Let also \rightarrow_s^* be the reflexive-transitive closure of \rightarrow_s . For every generalized ISPCF term M , and every open set U of normal forms, there is at most one enriched configuration (M', i', r') such that $(M, 0, 1) \rightarrow_s^* (M', i', r')$ and $M' \in U$. If it exists, we call that r' the *score* $\sigma[U](M, s)$ of M relative to s and U ; otherwise we let $\sigma[U](M, s) \stackrel{\text{def}}{=} 0$. Note that the score depends on M , on U , and on s .

One can refine this as follows. This will be needed below. Let $\rightarrow_s^{\leq k}$ denote reachability in at most k Next' steps.

Definition 12.2. For each $k \in \mathbb{N}$, let $\sigma_k[U](M, s)$ be the unique value r' such that $(M, 0, 1) \rightarrow_s^{\leq k} (M', i', r')$ for some $M' \in U$, if that exists, and 0 otherwise.

Then $\sigma[U]$ is the supremum of the directed family of maps $(\sigma_k[U])_{k \in \mathbb{N}}$.

LEMMA 12.3. For every $k \in \mathbb{N}$, for all $i, i'' \in \mathbb{N}$, $r, r' \in \mathbb{R}_+$, for all generalized ISPCF terms M and M' , $(M, i, r) \rightarrow_{\uparrow s}^{\leq k} (M', i', r')$ if and only if $(M, i + 1, r) \rightarrow_s^{\leq k} (M', i' + 1, r')$.

PROOF. It suffices to show that $(M, i, r) \rightarrow_{\uparrow s} (M', i', r')$ if and only if $(M, i + 1, r) \rightarrow_s (M', i' + 1, r')$, which is an easy verification, and to induct on k . □

The following is easily proved, too.

LEMMA 12.4. *For every $k \in \mathbb{N}$, for all $i, i'' \in \mathbb{N}$, $r, r' \in \mathbb{R}_+$, $\alpha \in \mathbb{R}_+$, for all generalized ISPCF terms M and M' , if $(M, i, 0) \rightarrow_s^{\leq k} (M', i', r')$ then $r' = 0$. If $(M, i, r) \rightarrow_s^{\leq k} (M', i', r')$ then $(M, i+1, \alpha.r) \rightarrow_s^{\leq k} (M', i'+1, \alpha.r')$. The converse implication holds if $\alpha \neq 0$.*

The proof of the following lemma, as well as all other missing proofs of this section, can be found in Appendix B.4.

LEMMA 12.5. *Under the assumptions of Definition 12.1, for every open subset U of Norm, the maps $\sigma_k[U]$ ($k \in \mathbb{N}$) and $\sigma[U]$ are lower semicontinuous from $\Gamma \times \{0, 1\}^{\mathbb{N}}$ to \mathbb{R}_+ .*

We define the ‘probability’ of reaching a normal form in some open subset U of Norm through the sampling-based operational semantics as:

$$P[M \rightarrow U] \stackrel{\text{def}}{=} \int_{s \in \{0, 1\}^{\mathbb{N}}} \sigma[U](M, s) \, dv.$$

In other words, $P[M \rightarrow U]$ is the average score of all strings of rewrite steps starting from M , where s is drawn uniformly at random. The integral makes sense by Lemma 12.5. We now verify that $P[M \rightarrow U]$ is exactly the ‘probability’ $\text{Next}^*(M)(U)$ that M eventually evaluates to a normal form in U according to the operational semantics of Section 10.

PROPOSITION 12.6. *Assume that all constants in Σ are first-order constants, and that the only available term of the form $\mathbf{sample}[\mu]$ is $\mathbf{sample}[\lambda_1]$, where λ_1 is Lebesgue measure on $[0, 1]$. For every open subset U of Norm, for every generalized ISPCF term M , $P[M \rightarrow U] = \text{Next}^*(M)(U)$.*

Combining Proposition 12.6 with Theorem 10.23, it follows that the denotational, operational, and sampling-based operational semantics all agree at observable types.

THEOREM 12.7. *Let Σ consist of strictly observable first-order constants, and let us assume that the only available term of the form $\mathbf{sample}[\mu]$ is $\mathbf{sample}[\lambda_1]$, where λ_1 is Lebesgue measure on $[0, 1]$. For every observable type β , for every generalized ISPCF term $M: D\beta$, for every observable open subset U of $\llbracket \beta \rrbracket$,*

$$\llbracket M \rrbracket (U) = \text{Next}^*(M)(\overline{U}) = P[M \rightarrow \overline{U}].$$

12.3 A few notes on implementation

We finish this section by briefly mentioning how the deterministic relation \rightarrow_s can be implemented. One only needs to be able to compute on exact real numbers. There are quite a number of proposals in order to do this [8, 13, 15, 19, 20, 39, 40, 53, 55, 58, 64, 78]. The only modification one needs to make is that some of the exact real numbers $\theta(x) \in \mathbb{R}$ we have to consider were drawn by sampling from $\mathbf{sample}[\lambda_1]$, i.e., were obtained as $\text{num}(s[i])$ for some $i \in \mathbb{N}$. Reading that number up to a specified accuracy $\epsilon > 0$ means reading its first k bits, where $1/2^k < \epsilon$. In an actual implementation, we would therefore only need (arbitrarily large) finite prefixes of an (arbitrarily large) finite number of substrings $s[i]$, and its bits can be drawn at random, and stored, on demand.

Explicitly, in the redundant signed-digit representation once proposed by Boehm and Cartwright in [8] and later by di Gianantonio and Lanzi [13], and by Ho [39], real numbers between -1 and 1 are infinite streams of digits in $\{-1, 0, 1\}$, to be read in base 2. Those infinite streams are read on demand, and can represent $s[i]$ verbatim. In Boehm and Cartwright’s second representation, real numbers x are encoded as functions \bar{x} mapping each natural number k to an integer $\bar{x}(k)$ such that $\bar{x}(k)/4^k$ is at distance at most $1/4^k$ from x . One can then represent $s[i]$ by maintaining a table

T_i of the first few bits of $s[i]$ in mutable store. When $\overline{s[i]}(k)$ is required, we make sure that at least the first $2k$ entries of T_i are populated, possibly drawing fresh bits, uniformly and independently at random, and storing them in T_i ; then we return $\sum_{j=0}^{2k-1} T_i[j]/2^{j+1}$.

The other primitives working on real numbers that we have considered are well covered in the literature, except perhaps for bin' and num' , mentioned in Section 9.5. We have already described how bin' can be implemented, right before Remark 9.1. In a redundant signed-digit representation, Ho explains how one can compute series of the form $\sum_{k=0}^{+\infty} x_k/2^{k+1}$, where $(x_0, x_1, \dots, x_k, \dots)$ is a stream of exact real numbers in $[-1, 1]$ [39, Section 3.1.2], based on unpublished work by Adam Scriven. (This is also the denotation of Simpson's coerce operation [74, Figure 3].) From this, num' is easily implemented. In Boehm and Cartwright's second representation, one can implement $\text{num}'(s)$ as the function that maps each $k \in \mathbb{N}$ to the partial, finite sum $\sum_{k'=0}^{2k-1} s_{k'}/2^{k'+1}$ if s_0, \dots, s_{2k-1} are all different from \perp , and as \perp otherwise. While this definition of num' differs slightly from (4), it shares with it the property that $\text{num}'(s)$ computes the exact real representation of $\text{num}(s)$ if no bit s_k is equal to \perp , which is the only thing we require from it.

13 RELATED WORK

We have already cited a few papers on statistical programming languages [29, 30, 57, 85] and, what is of more interest to us, on *semantics* of such languages [10, 17, 38, 81]. We will not cite the even larger body of literature that deals with the simpler case of probabilistic, higher-order languages without real numbers, continuous distributions, or soft constraints. We mention [62], which presents a sampling-based operational semantics for a higher-order probabilistic call-by-value language with continuous distributions and true real numbers, and which can be seen as a precursor of some of the previously mentioned papers.

We have insisted on obtaining a commutative monad of continuous R -valuations, and we have noted that this amounts to forms of the Fubini-Tonelli theorem. There are several such commutative monads in the literature already, including some that predate the novel commutative monad of [81]. On the category of measurable spaces, the monad of spaces of probability measures, popularized by M. Giry [28], is commutative, as first proved by L. Tonelli [80]. (Tonelli's theorem applies to non-negative measurable maps, while the Fubini-Tonelli theorem applies to measurable maps whose absolute value is integrable.) Vickers introduced a valuation monad on the category of locales, and proved that it is commutative, too [82]. The topological counterpart of that result is Proposition 4.1. We have argued in Section 4 why this does not imply a corresponding Fubini-Tonelli theorem on the category \mathbf{Dcpo} of dcpo s.

We get around this problem by considering *minimal* valuations instead of general continuous valuations. The minimal *subprobability* valuations are an instance of a more general construction of Jia and Mislove [43], based on Keimel and Lawson's K -completions [49]. Heckmann's point-continuous valuations [36] are another instance of that construction. That the minimal valuations provide for commutative monads on \mathbf{Dcpo} was also independently realized by the first author in 2020. This was discovered during a collaboration between the first two authors when preparing a paper submitted at the LICS'21 conference, which eventually leads to this work. The minimal valuations monad was also applied to the semantics of higher-order probabilistic programs by Jia, Lindenhovius, Mislove, and Zamdzhiev [42], but with only discrete basic types, namely without a type of reals.

The general idea of using a form of completion of simple objects (simple valuations, here) in order to obtain objects with better properties is, naturally, a very common idea. A recent and topical example is the construction of the commutative monad T of [81, Section 4]. There, the authors consider the continuation monad J with answers in $[0, +\infty]$. The analogue in our case would be to define JX as $[[X \rightarrow \overline{\mathbb{R}}_+] \rightarrow \overline{\mathbb{R}}_+] = [\mathcal{L}X \rightarrow \overline{\mathbb{R}}_+]$, and one should realize that VX embeds into JX by mapping each continuous valuation ν to its integration functional $\int _ dv$. Then, they consider

the randomisable expectation operators, which are obtained as the (integration functionals of) the image measures of Lebesgue measure on \mathbb{R} . Finally, T is the smallest full submonad of J that contains all the randomisable expectation operators. A simpler, and more classical example of completion is the notion of *s-finite* measure [26, 73], which are the suprema of countable chains of bounded measures; s-finite measures satisfy Fubini-Tonelli, but are not known to form a monad on the category \mathbf{Meas} of measurable sets and measurable maps. Instead, s-finite kernels form a commutative monad on \mathbf{Meas} , which Staton used to give semantics to a simple first-order probabilistic language [75].

14 CONCLUSION

We have given a domain-theoretic, denotational semantics for a statistical programming language, based on the novel notion of *minimal* valuation, forming a commutative monad over \mathbf{Dcpo} . We believe that the notion is natural, and that the resulting denotational semantics is simple. Through an extensive list of examples, we hope to have demonstrated that a language based on this semantics is able to implement a rich set of distributions, including some non-trivial distributions on higher-order objects. We have given three operational semantics, which are all sound and adequate, under various, natural sets of assumptions on the semantics of available constants. Further work include formally verified efficient implementations, for example extending the particle Monte Carlo Markov chain algorithm of Anglican [85].

ACKNOWLEDGMENTS

We thank the anonymous referees for their comments. We also took some advice from the anonymous referees on an earlier, and very different version of this paper submitted at the LICS'21 conference. We thank Vladimir Zamdzhiev, who spotted a mistake in the claim of soundness in one version of this paper. We thank Daniel Roy for mentioning theories of computable probability distributions to us, and Sam Staton for correcting a misunderstanding we had about s-finite measures. We have had a long and enlightening discussion with Ohad Kammar about the relation of our construction of \mathbf{V}_m with the construction of [48]. *Remark 5.3 should have been a short summary of it. But the issues were so many and subtle; the remark started to spiral down into a long list of failed attempts, and complex issues, which were certainly at least as hard as, if not much harder than, building \mathbf{V}_m directly. Hence almost nothing remains of our discussions with Ohad Kammar, or with the anonymous referees, on the subject, but it is fair to say that we learned a lot from them.* Xiaodong Jia acknowledges support from NSFC (No. 12231007, No. 12001181).

REFERENCES

- [1] Samson Abramsky and Achim Jung. 1994. Domain Theory. In *Semantic Structures*, Samson Abramsky, Dov M. Gabbay, and Thomas Stephen Edward Maibaum (Eds.). Handbook of Logic in Computer Science, Vol. 3. Clarendon Press, Oxford, 1–168.
- [2] Nathanael Leedom Ackerman, Cameron Eric Freer, and Daniel Murphy Roy. 2011. Noncomputable Conditional Distributions. In *Proc. 26th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'11)*. IEEE, Toronto, ON, Canada, 107–116. <https://doi.org/10.1109/LICS.2011.49>.
- [3] Wolfgang Adamski. 1977. τ -Smooth Borel Measures on Topological Spaces. *Mathematische Nachrichten* 78 (1977), 97–107.
- [4] M. Alvarez-Manilla. 2002. Extension of valuations on locally compact sober spaces. *Topology and its Applications* 124 (2002), 397–433.
- [5] Mauricio Alvarez-Manilla, Abbas Edalat, and Nasser Saheb-Djahromi. 2000. An Extension Result for Continuous Valuations. *Journal of the London Mathematical Society* 61 (2000), 629–640.
- [6] Patrick Billingsley. 1986. *Probability and Measure* (3rd, 1995 ed.). John Wiley and Sons, New York, Chichester, Weinheim, Brisbane, Singapore, Toronto.
- [7] Richard Bird. 1998. *Introduction to Functional Programming using Haskell* (2nd ed.). Prentice-Hall Series in Computer Science, Prentice-Hall Europe, 433 pages.
- [8] Hans-Joachim Boehm, Robert Cartwright, Mark Riggall, and Michael James O'Donnell. 1986. Exact Real Arithmetic: A Case Study in Higher Order Programming. In *Proc. ACM conference on LISP and functional programming*. ACM Press, Cambridge, Massachusetts, USA, 162–173.

- [9] George Edward Pelham Box and Mervin Edgar Muller. 1958. A Note on the Generation of Random Normal Deviates. *The Annals of Mathematical Statistics* 29, 2 (1958), 610–611. <https://doi.org/10.1214/aoms/1177706645>
- [10] Ugo Dal Lago and Naohiko Hoshino. 2019. The Geometry of Bayesian Programming. In *34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'19)*. IEEE, Vancouver Canada, 1–13. <https://doi.org/10.1109/LICS.2019.8785663>
- [11] Matthew de Brecht. 2013. Quasi-Polish Spaces. *Annals of Pure and Applied Logic* 164, 3 (2013), 356–381.
- [12] Matthew de Brecht, Jean Goubault-Larrecq, Xiaodong Jia, and Zhenchao Lyu. 2019. Domain-Complete and LCS-Complete Spaces. In *Proceedings of the International Symposium on Domain Theory (ISDT'19) (Electronic Notes in Theoretical Computer Science, Vol. 345)*, Achim Jung, Qingguo Li, Luoshan Xu, and Guo-Qiang Zhang (Eds.). Elsevier Science Publishers, Yangzhou, China, 3–35. <https://doi.org/10.1016/j.entcs.2019.07.014>
- [13] Pietro Di Gianantonio and Pier Luca Lanzi. 2004. Algorithms for Exact Real Arithmetic. *Electronic Notes in Theoretical Computer Science* 104 (2004), 113–128.
- [14] Abbas Edalat and Martín Hötzel Escardó. 2000. Integration in Real PCF. *Information and Computation* 160 (2000), 128–166. <https://doi.org/10.1006/inco.1999.2844>
- [15] Abbas Edalat, Martín Hötzel Escardó, and Peter Potts. 1997. Semantics of Exact Real Arithmetic. In *Proc. 12th Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society, Warsaw, Poland, 248–257.
- [16] Abbas Edalat and Sara Negri. 1998. The Generalized Riemann Integral on Locally Compact Spaces. *Topology and its Applications* 89, 1–2 (1998), 121–150. [https://doi.org/10.1016/S0166-8641\(97\)00227-7](https://doi.org/10.1016/S0166-8641(97)00227-7)
- [17] Thomas Ehrhard, Michele Pagani, and Christine Tasson. 2018. Measurable Cones and Stable, Measurable Functions: A Model for Probabilistic Higher-Order Programming. In *Proc. 45th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL'18)*. Association for Computing Machinery, Los Angeles, US, 1–28. <https://doi.org/10.1145/3158147> Article 59.
- [18] Yuri Leonidovich Ershov. 1997. The Bounded Complete Hull of an α -Space. *Theoretical Computer Science* 175 (1997), 3–13.
- [19] Martín Hötzel Escardó. 1996. PCF Extended with Real Numbers. *Theoretical Computer Science* 162 (1996), 79–115.
- [20] Martín Hötzel Escardó. 1996. *PCF extended with real numbers: a domain-theoretic approach to higher-order exact real number computation*. Ph. D. Dissertation. Imperial College, London, UK.
- [21] Baoying Fan, Yusong Du, Baodian Wei, and Xiao Ma. 2019. On Generating Exponentially Distributed Variates by Using Early Rejection. In *5th IEEE International Conference on Computer and Communications*. IEEE, Patna, India, 1307–1311.
- [22] Thomas Ferguson. 1973. Bayesian Analysis of some Nonparametric Problems. *Annals of Statistics* 1, 2 (1973), 209–230. <https://doi.org/10.1214/aos/1176342360>
- [23] Marcelo P. Fiore and Gordon D. Plotkin. 1994. An Axiomatisation of Computationally Adequate Domain Theoretic Models of FPC. In *Proc. 9th Annual IEEE Symposium on Logic in Computer Science (LICS'94)*. IEEE, IEEE, Paris, France, 92–102.
- [24] George Elmer Forsythe. 1972. Von Neumann's Comparison Method for Random Sampling from the Normal and Other Distributions. *Math. Comp.* 26, 120 (1972), 817–826.
- [25] Stefano Galatolo, Mathieu Hoyrup, and Cristóbal Rojas. 2010. Effective Symbolic Dynamics, Random Points, Statistical Behavior, Complexity and Entropy. *Information and Computation* 208, 1 (2010), 23–41.
- [26] Ronald Kay Getoor. 1990. *Excessive Measures*. Birkhäuser, Basel, Switzerland.
- [27] Gerhard Gierz, Karl Heinrich Hofmann, Klaus Keimel, Jimmie D. Lawson, Michael Mislove, and Dana Stewart Scott. 2003. *Continuous Lattices and Domains*. Encyclopedia of Mathematics and its Applications, Vol. 93. Cambridge University Press, Cambridge.
- [28] Michèle Giry. 1982. A Categorical Approach to Probability Theory. *Categorical aspects of topology and analysis* 915 (January 1982), 68–85.
- [29] Noah Goodman, Vikash Mansinghka, Daniel Murphy Roy, Keith Bonawitz, and Joshua Brett Tenenbaum. 2008. Church: A Language for Generative Models. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI'08)*. AUAI Press, Helsinki, Finland, 220–229.
- [30] Noah D. Goodman and Andreas Stuhlmüller. 2014. The Design and Implementation of Probabilistic Programming Languages. <http://dippl.org>. Accessed: 2021-10-1.
- [31] Jean Goubault-Larrecq. 2013. *Non-Hausdorff Topology and Domain Theory*. New Mathematical Monographs, Vol. 22. Cambridge University Press, Cambridge, UK.
- [32] Jean Goubault-Larrecq. 2015. Full Abstraction for Non-Deterministic and Probabilistic Extensions of PCF I: the Angelic Cases. *Journal of Logic and Algebraic Methods in Programming* 84, 1 (Jan. 2015), 155–184. <https://doi.org/10.1016/j.jlamp.2014.09.003>
- [33] Jean Goubault-Larrecq. 2019. A Probabilistic and Non-Deterministic Call-by-Push-Value Language. In *Proc. 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'19)*, Patricia Bouyer (Ed.). IEEE Press, Vancouver, Canada, 1–13. <https://doi.org/10.1109/LICS.2019.8785809>
- [34] Jean Goubault-Larrecq. 2021. Products and Projective Limits of Continuous Valuations on T_0 Spaces. *Mathematical Structures in Computer Science* 31 (2021), 1–21. <https://doi.org/10.1017/S0960129521000141>
- [35] Jean Goubault-Larrecq and Xiaodong Jia. 2021. Separating minimal valuations, point-continuous valuations, and continuous valuations. *Mathematical Structures in Computer Science* 31, 6 (2021), 614–632. <https://doi.org/10.1017/S0960129521000384>
- [36] Reinhold Heckmann. 1995. *Spaces of Valuations*. Technical Report A 09/95. FB 14 Informatik, Universität des Saarlandes, 66041 Saarbrücken, Germany.
- [37] Reinhold Heckmann. 1996. Spaces of Valuations. In *Papers on General Topology and Applications: Eleventh Summer Conference at the University of Southern Maine (Annals of the New York Academy of Sciences, Vol. 806)*, Susan Andima, Robert C. Flagg, Gerald Itzkowitz, Ralph Kopperman, Yung Kong, and Prabudh Misra (Eds.). New York Academy of Sciences, New York, US, 174–200.

- [38] Chris Heunen, Ohad Kammar, Sam Staton, and Hongseok Yang. 2017. A Convenient Category for Higher-Order Probability Theory. In *Proc. 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'17)*. ACM, Reykjavik Iceland, 1–12.
- [39] Weng Kin Ho. 2013. Exact Real Calculator for Everyone. In *Proceedings of the 1st Asian Technology Conference in Mathematics (ATCM'13)*, W.-C. Yang, M. Majweski, T. Alwis, and I. K. Rana (Eds.). Mathematics and Technology, LLC, Bombay, India, 1–15.
- [40] Daniel Huang, Greg Morrisett, and Bas Spitters. 2020. Application of Computable Distributions to the Semantics of Probabilistic Programs. In *Foundations of Probabilistic Programming*, Gilles Barthe, Joost-Pieter Katoen, and Alexandra Silva (Eds.). Cambridge University Press, Cambridge, UK, 75–120. <https://doi.org/10.1017/9781108770750.004>
- [41] John Isbell. 1981. Product Spaces in Locales. *Proc. Amer. Math. Soc.* 81, 1 (1981), 116–118.
- [42] Xiaodong Jia, Bert Lindenhovius, Michael Mislove, and Vladimir Zamdzhiev. 2021. Commutative Monads for Probabilistic Programming Languages. In *Proc. 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'21)*. IEEE Press, Rome, Italy, 1–14. <https://doi.org/10.1109/LICS52264.2021.9470611> Article 19.
- [43] Xiaodong Jia and Michael Mislove. 2022. Completing simple valuations in K-categories. *Topology and its Applications* 318 (2022), 108192. <https://doi.org/10.1016/j.topol.2022.108192>
- [44] Claire Jones. 1990. *Probabilistic Non-Determinism*. Ph.D. Dissertation. University of Edinburgh, Edinburgh. Also published as Technical Report No. CST-63-90.
- [45] Claire Jones and Gordon Plotkin. 1989. A probabilistic powerdomain of evaluations. In *Proceedings of the 4th Annual Symposium on Logic in Computer Science*. IEEE Press, California, USA, 186–195.
- [46] Achim Jung and Regina Tix. 1998. The Troublesome Probabilistic Powerdomain. In *Proceedings of the Third Workshop on Computation and Approximation (Electronic Notes in Theoretical Computer Science, Vol. 13)*, Abbas Edalat, Achim Jung, Klaus Keimel, and Marta Kwiatkowska (Eds.). Springer-Verlag, Berlin, Heidelberg, 70–91. 23 pages.
- [47] Olav Kallenberg. 2002. *Foundations of Modern Probability* (2nd edition ed.). Springer-Verlag, New York, US.
- [48] Ohad Kammar and Dylan McDermott. 2018. Factorisation Systems for Logical Relations and Monadic Lifting in Type-and-effect System Semantics. *Electronic Notes in Theoretical Computer Science* 341 (2018), 239–260. <https://doi.org/10.1016/j.entcs.2018.11.012> Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics (MFPS XXXIV).
- [49] Klaus Keimel and Jimmie D. Lawson. 2008. D-completions and the d -topology. *Annals of Pure and Applied Logic* 159 (2008), 292–306. <https://doi.org/10.1016/j.apal.2008.06.019>
- [50] Olaf Kirch. 1993. *Bereiche und Bewertungen*. Master's thesis. Technische Hochschule Darmstadt. 77 pages.
- [51] Anders Kock. 1971. Bilinearity and Cartesian Closed Monads. *Math. Scand.* 29 (1971), 161–174.
- [52] Anders Kock. 2012. Commutative Monads as a Theory of Distributions. *Theory and Applications of Categories* 26, 4 (2012), 97–131.
- [53] Branimir Lambov. 2005. Complexity in a type-1 framework for computable analysis. In *CiE 2005: New Computational Paradigms (ILLC Technical Notes (X) Series)*, S. B. Cooper, B. Löwe, and L. Torenvliet (Eds.). Institute for Logic, Language, and Computation, Springer, Amsterdam, 133–145.
- [54] Jimmie D. Lawson. 1997. Spaces of maximal points. *Mathematical Structures in Computer Science* 7 (1997), 543–555.
- [55] David R. Lester. 1992. Vuillemin's Exact Real Arithmetic. In *Functional Programming Workshop, Glasgow 1991 (BCS Workshops in Computing)*, Rogardt Hødal, Carsten Kehler Holst, and Philip Wadler (Eds.). Springer Verlag, Glasgow, 225–238.
- [56] Ernest G. Manes. 1976. *Algebraic Theories*. Graduate Texts in Mathematics, Vol. 26. Springer-Verlag, New York, US.
- [57] Vikash K. Mansinghka, Daniel Selsam, and Yura N. Perov. 2014. Venture: A Higher-Order Probabilistic Programming Platform with Programmable Inference. arXiv report, <http://arxiv.org/abs/1404.0099>.
- [58] José Raymundo Marcial-Romero. 2004. *Semantics of a sequential language for exact real-number computation*. Ph.D. Dissertation. University of Birmingham.
- [59] Eugenio Moggi. 1989. Computational Lambda-Calculus and Monads. In *Logic in Computer Science*. IEEE Computer Society Press, California, USA, 14–23.
- [60] Eugenio Moggi. 1991. Notions of Computations and Monads. *Information and Computation* 93, 1 (1991), 55–92.
- [61] Praveen Narayanan, Jacques Carette, Wren Romano, Chung-Chieh Shan, and Robert Zinkov. 2016. Probabilistic Inference by Program Transformation in Hakaru. In *Proc. 13th Intl. Symp. on Functional and Logic Programming (FOOPS'16)*. Springer, Kochi, Japan, 62–79. System description.
- [62] Sungwoo Park, Frank Pfenning, and Sebastian Thrun. 2008. A Probabilistic Language Based on Sampling Functions. *ACM Transactions on Programming Languages and Systems* 31, 1 (2008), 1–46. <https://doi.org/10.1145/1452044.1452048> Article 4.
- [63] Andrew M. Pitts. 1993. *Relational Properties of Domains*. Technical Report 321. Cambridge Univ. Computer Laboratory. 37 pages.
- [64] David Plume. 1998. A Calculator for Exact Real Number Computation. <http://www.dcs.ed.ac.uk/home/mhe/plume/index.html>. 4th year project, Dpt. of Computer Science and Artificial Intelligence, University of Edinburgh.
- [65] Daniel Murphy Roy. 2011. *Computability, Inference and Modeling in Probabilistic Programming*. Ph.D. Dissertation. MIT.
- [66] Nasser Saheb-Djahromi. 1980. CPO's of measures for nondeterminism. *Theoretical Computer Science* 12 (1980), 19–37.
- [67] René L. Schilling. 2017. *Measures, Integrals and Martingales* (second edition ed.). Measures, Integrals and Martingales, Cambridge, New York, Port Melbourne, Delhi, Singapore.
- [68] Matthias Schröder and Alex Simpson. 2006. Representing Probability Measures using Probabilistic Processes. *Journal of Complexity* 22, 6 (2006), 768–782.
- [69] Holger Schulz. 2000. Type Two Theory of Effectivity and Real PCF. *Electronic Notes in Theoretical Computer Science* 35 (2000), 191–203.

- [70] Dana Stewart Scott. 1969. A type theoretic alternative to ISWIM, CUCH, OWHY. (1969). Manuscript, University of Oxford. First published as [71].
- [71] Dana Stewart Scott. 1993. A type theoretic alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science* 121 (1993), 411–440.
- [72] Chung-Chieh Shan and Norman Ramsey. 2017. Exact Bayesian Inference by Symbolic Disintegration. In *Proc. 44th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL’17)*. Association for Computing Machinery, Paris, France, 130–144.
- [73] Michael Sharpe. 1988. *General Theory of Markov Processes*. Academic Press, Cambridge.
- [74] Alex Simpson. 1998. Lazy functional algorithms for exact real functionals. In *Mathematical Foundations of Computer Science 1998 (Lecture Notes in Computer Science, Vol. 1450)*. Springer Verlag, Brno, Czech Republic, 456–464.
- [75] Sam Staton. 2017. Commutative Semantics for Probabilistic Programming. In *Proc. 26th European Symposium on Programming (ESOP’17)*, Hongseok Yang (Ed.). Springer Verlag Lecture Notes in Computer Science 10201, Uppsala, Sweden, 855–879. https://doi.org/10.1007/978-3-662-54434-1_32
- [76] Sam Staton, Hongseok Yang, Chris Heunen, Ohad Kammar, and Frank Wood. 2016. Semantics for Probabilistic Programming: Higher-Order Functions, Continuous Distributions, and Soft Constraints. In *Proc. 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS’16)*. ACM, New York, US, 525–534.
- [77] Thomas Streicher. 2002. Mathematical Foundations of Functional Programming. (2002). Available from <http://www.mathematik.tu-darmstadt.de/~streicher/MGFP/mgfp.ps.gz>.
- [78] Philipp Sünderhauf. 1995. A faithful computational model of the real numbers. *Theoretical Computer Science* 151, 1 (1995), 277–294. [https://doi.org/10.1016/0304-3975\(95\)00054-Z](https://doi.org/10.1016/0304-3975(95)00054-Z) Selected Papers of the Workshop on Topology and Completion in Semantics.
- [79] Regina Tix. 1995. *Stetige Bewertungen auf topologischen Räumen*. Master’s thesis. Technische Hochschule Darmstadt. 51 pages.
- [80] Leonida Tonelli. 1909. Sull’integrazione per Parti. *Atti della Accademia Nazionale dei Lincei* 18, 2 (1909), 246–253.
- [81] Matthijs Vákár, Ohad Kammar, and Sam Staton. 2019. A Domain Theory for Statistical Probabilistic Programming. *Proc. 46th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL’19)* 3 (2019), 1–29. <https://doi.org/10.1145/3290349> Article 36.
- [82] Steven J. Vickers. 2011. A Monad of Valuation Locales. <https://www.cs.bham.ac.uk/~sjv/Riesz.pdf>.
- [83] Klaus Weihrauch. 1987. *Computability*. Springer Verlag, Berlin Heidelberg.
- [84] Klaus Weihrauch. 1999. Computability on the Probability Measures on the Borel Sets of the Unit Interval. *Theoretical Computer Science* 219 (1999), 421–437.
- [85] Frank Wood, Jan Willem van de Meent, and Vikash Mansinghka. 2014. A New Approach to Probabilistic Programming Inference. In *Proceedings of the 17th International Conference on Artificial Intelligence and Statistics (AISTATS’14) (Proceedings of Machine Learning Research, Vol. 33)*, Samuel Kaski and Jukka Corander (Eds.). PMLR, Reykjavik, Iceland, 1024–1032.

A EXAMPLES CONTINUED

A.1 The call-by-value versus call-by-name question

A recurring paradox in the theory of higher-order probabilistic programming languages is the following: what should the value of $(\lambda x. x = x)(\text{rand}())$ be, where $\text{rand}()$ returns a random number? Since $x = x$ is universally true, one expects this program to always return true. In call-by-name, one also expects β -reduction to be a sound reduction rule, so that this value should also be equal to $\text{rand}() = \text{rand}()$. Unfortunately, the latter draws *two* independent random numbers, and may return false.

This problem is avoided in *call-by-value* languages, which force $\text{rand}()$ to be evaluated first, and only once.

This need not be a problem in call-by-name languages such as ISPCF. This is well-known (ISPCF is a call-by-name language with a monad, just like Haskell [7]), but we would like this point to be clear. In the call-by-name language PPCF, the issue is solved by using a $\text{let}(x, M, N)$ construct that evaluates M , binds the resulting value to x and then evaluates N [17, Example 3.10]. In ISPCF, **do** may serve a similar purpose, as we now demonstrate.

Let us build a simple uniform random number generator randbool on **bool**. We recall that **bool** $\stackrel{\text{def}}{=} \text{unit} + \text{unit}$. We define **true** as ι_1* ; its semantics is $\text{true} = (1, *)$; and **false** as ι_2* , whose semantics is $\text{false} = (2, *)$. If M has type **bool**, we also define **if** MNP as an abbreviation for **case** $M(\lambda x_1. N)(\lambda x_2. P)$, where x_1 and x_2 are fresh variables.

In order to implement randbool , we assume we have a term $\text{sample}[\lambda_1]$, where λ_1 is the uniform measure on $[0, 1]$, namely: for every measurable subset E of \mathbb{R} , $\lambda_1(E) \stackrel{\text{def}}{=} \lambda(E \cap [0, 1])$. We also assume the pos primitive, and subtraction

–. We define `randbool` as follows.

$$\text{randbool} \stackrel{\text{def}}{=} \mathbf{do} \ x_{\text{real}} \leftarrow \mathbf{sample}[\lambda_1]; \mathbf{ret}(\text{pos}(x - 0.5))$$

For every environment ρ , $\llbracket \text{randbool} \rrbracket \rho$ is the continuous valuation $\frac{1}{2}\delta_{\text{true}} + \frac{1}{2}\delta_{\text{false}}$.

Let `eqbool` $\stackrel{\text{def}}{=} \lambda x.\lambda y.\mathbf{if} \ x \ (\mathbf{if} \ y \ \text{true} \ \text{false}) \ (\mathbf{if} \ y \ \text{false} \ \text{true})$, and we abbreviate `eqbool` MN as $M = N$, for all $M, N: \mathbf{bool}$. One cannot write $(\lambda x_{\mathbf{bool}}.x = x)(\text{randbool})$ directly, because `randbool` has type $D\mathbf{bool}$, not \mathbf{bool} . Sampling from a distribution must be done through `do`, and we have (at least) two choices. The term:

$$\mathbf{do} \ b_{\mathbf{bool}} \leftarrow \text{randbool}; \mathbf{ret}((\lambda x_{\mathbf{bool}}.x = x)b)$$

first samples, obtaining a random Boolean value b , which is then passed to $\lambda x_{\mathbf{bool}}.x = x$. This term has semantics δ_{true} ; in other words, it is a probabilistic process that returns true with probability one. Or we may defer sampling as follows:

$$(\lambda x_{D\mathbf{bool}}.\mathbf{do} \ b_1 \leftarrow x; \mathbf{do} \ b_2 \leftarrow x; \mathbf{ret}(b_1 = b_2))\text{randbool}.$$

The semantics of that term is $\frac{1}{2}\delta_{\text{true}} + \frac{1}{2}\delta_{\text{false}}$: the probability that the two independent Boolean values b_1 and b_2 are equal is only 1/2.

A.2 Verifying that Lebesgue implements Lebesgue valuation on \mathbb{R}

We verify that `lebesgue` implements Lebesgue valuation correctly, formally, here. We wish to show that $\llbracket \text{lebesgue} \rrbracket \rho = i[\lambda]$.

First, for every environment ρ , for all $a, b \in \mathbb{R}$ such that $a < b$, $\llbracket \text{uniform} \rrbracket \rho(a)(b)$ is the minimal valuation $(\lambda x.\delta_{a+(b-a)x})^\dagger(i[\lambda_1])$. For every open subset U of \mathbb{IR}_\perp , therefore,

$$\begin{aligned} \llbracket \text{uniform} \rrbracket \rho(a)(b)(U) &= \int_{\mathbf{u} \in \mathbb{IR}_\perp} \delta_{i(a)+(i(b)-i(a))\times \mathbf{u}}(U) \, di[\lambda_1] \\ &= \int_{u \in \mathbb{R}} \delta_{(i(a)+(i(b)-i(a))\times i(u))}(U) \, d\lambda_1 \\ &\text{by the change-of-variables formula} \\ &= \int_{u \in \mathbb{R}} \delta_{i(a+(b-a)u)}(U) \, d\lambda_1 \\ &= \int_0^1 \chi_{i^{-1}(U)}(a + (b-a)u) \, du \\ &\text{using the identity } \delta_c(U) = \chi_U(c) \\ &= \frac{1}{b-a} \int_a^b \chi_{i^{-1}(U)}(v) \, dv \\ &\text{by letting } v \stackrel{\text{def}}{=} a + (b-a)u \\ &= v_{[a,b]}(i^{-1}(U)). \end{aligned}$$

Therefore $\llbracket \text{uniform} \rrbracket \rho(a)(b) = i[v_{[a,b]}]$.

Next, we evaluate $\llbracket \text{lebesgue} \rrbracket \rho(U)$ for every open subset U of \mathbb{IR}_\perp as follows. Let us abbreviate the term `score(1.0/(1.0+z x z)); ret(arctan z)` as $M(z)$. For every $u \in \mathbb{R}$, for every open subset U of \mathbb{IR}_\perp , letting $v_u \stackrel{\text{def}}{=} \llbracket M(z) \rrbracket \rho[z \mapsto i(u)]$, we

have the following, where $g(u) \stackrel{\text{def}}{=} \arctan u$, and therefore $g'(u) = 1/(1 + u^2)$:

$$\begin{aligned} v_u(U) &= (\lambda_{\cdot} \delta_{g(u)})^\dagger (g'(u) \delta_*)(U) \\ &= \int_{\cdot \in \llbracket \mathbf{unit} \rrbracket} \delta_{g(u)}(U) dg'(u) \delta_* \\ &= g'(u) \chi_{i^{-1}(U)}(g(u)). \end{aligned}$$

Therefore,

$$\begin{aligned} \llbracket \text{lebesgue} \rrbracket \rho(U) &= \int_{\mathbf{u} \in \mathbb{R}_\perp} \llbracket M(z) \rrbracket \rho[z \mapsto \mathbf{u}](U) di[v_{[-\pi/2, \pi/2]}] \\ &= \int_{u \in \mathbb{R}} \llbracket M(z) \rrbracket \rho[z \mapsto i(u)](U) dv_{[-\pi/2, \pi/2]} \\ &\text{by the change-of-variables formula} \\ &= \int_{u \in \mathbb{R}} v_u(U) dv_{[-\pi/2, \pi/2]} \\ &= \int_{u \in \mathbb{R}} g'(u) \chi_{i^{-1}(U)}(g(u)) dv_{[-\pi/2, \pi/2]} \\ &= \int_{-\pi/2}^{\pi/2} g'(u) \chi_{i^{-1}(U)}(g(u)) du \\ &= \int_{-\infty}^{+\infty} \chi_{i^{-1}(U)}(t) dt \\ &\text{by letting } t \stackrel{\text{def}}{=} g(u) \\ &= \lambda(i^{-1}(U)). \end{aligned}$$

Therefore, as promised, $\llbracket \text{lebesgue} \rrbracket \rho$ is the image **valuation** $i[\lambda]$ of Lebesgue **valuation** λ on \mathbb{R} .

A.3 Other implementations of the exponential distribution

Another way of implementing the exponential distribution is as:

$$\text{expo}' \stackrel{\text{def}}{=} \mathbf{do } x_{\text{real}} \leftarrow \mathbf{sample}[\lambda_1]; \mathbf{ret}(-\log x),$$

assuming a term $\mathbf{sample}[\lambda_1]$ for sampling along Lebesgue measure λ_1 on $[0, 1]$. We check that this also gives the correct result. For every open subset W of \mathbb{R}_\perp ,

$$\begin{aligned} \llbracket \text{expo}' \rrbracket \rho(W) &= \int_{\mathbf{x} \in \mathbb{R}_\perp} \delta_{-\llbracket \log \rrbracket} \rho(\mathbf{x})(W) di[\lambda_1] \\ &= \int_0^1 \chi_W(-\log x) dx, \end{aligned}$$

by the change-of-variables formula, the fact that integrating along λ_1 means integrating on $[0, 1]$, and the equality $\delta_a(W) = \chi_W(a)$. We now let $x \stackrel{\text{def}}{=} e^{-t}$. The latter integral is then equal to $\int_0^{+\infty} \chi_W(t) e^{-t} dt = (g \cdot \lambda)(W)$. Therefore $\llbracket \text{expo}' \rrbracket \rho = \llbracket \text{expo} \rrbracket \rho = i[g \cdot \lambda]$.

A final implementation of the exponential distribution, due to von Neumann, consists in simulating a distribution with density $(1 - x) + (\frac{x^2}{2!} - \frac{x^3}{3!}) + \dots + (\frac{x^{2n}}{(2n)!} - \frac{x^{2n+1}}{(2n+1)!}) + \dots = e^{-x}$. We will leave the verification that the term

von_neumann below again has the same semantics $i[g \cdot \lambda]$ as expo and expo', as an exercise. We first define:

$$\begin{aligned} \text{longest_decreasing_run} &\stackrel{\text{def}}{=} \mathbf{rec}(\lambda f_{\mathbf{real} \rightarrow \mathbf{int} \rightarrow D\mathbf{int}} . \lambda x_{\mathbf{real}} . \lambda n_{\mathbf{int}} . \\ &\quad \mathbf{do} \ u \leftarrow \mathbf{sample}[\lambda_1]; \\ &\quad \mathbf{if} \ (\underline{\text{pos}}(u - x)) \\ &\quad \quad \mathbf{ret} \ n \\ &\quad \quad (f \ u \ (n + \underline{1}))) \end{aligned}$$

Given any $x \in [0, 1]$, $\llbracket \text{longest_decreasing_run} \rrbracket \rho(x)(0)$ returns the largest n such that $x > u_1 > u_2 > \dots > u_n$ ($\leq u_{n+1}$) for randomly uniformly distributed real numbers u_1, u_2, \dots , in $[0, 1]$. It is a standard exercise to show that $\llbracket \text{longest_decreasing_run} \rrbracket \rho(x)(0)$ is the distribution on \mathbb{Z}_{\perp} giving probability $\frac{x^n}{n!} - \frac{x^{n+1}}{(n+1)!}$ to each $n \in \mathbb{N}$, and probability 0 to all other elements. Drawing a value n at random with respect to that distribution, the probability that n is even is $\sum_{n \text{ even}} (\frac{x^n}{n!} - \frac{x^{n+1}}{(n+1)!}) = e^{-x}$. Assuming a term $\underline{\text{odd}}: \mathbf{int} \rightarrow \mathbf{bool}$ testing whether its argument is odd, one then defines:

$$\begin{aligned} \text{von_neumann} &\stackrel{\text{def}}{=} \mathbf{do} \ x_{\mathbf{real}} \leftarrow \mathbf{sample}[\lambda_1]; \\ &\quad \mathbf{rec}(\lambda f_{\mathbf{real} \rightarrow D\mathbf{real}} . \lambda \ell_{\mathbf{real}} . \\ &\quad \quad \mathbf{do} \ b_{\mathbf{bool}} \leftarrow \text{longest_decreasing_run } x \ 0; \\ &\quad \quad \mathbf{if} \ b \\ &\quad \quad \quad f \ (\ell + \underline{1.0}) \\ &\quad \quad \mathbf{ret} \ \ell) \ x \end{aligned}$$

The **rec** expression in the middle maps every real number ℓ to $\ell + m$, where m is the average number of calls to $\text{longest_decreasing_run } x \ 0$ it takes it to return an even number.

We direct the reader to [21] for an explanation and some faster algorithms based on the same principle. See also [24] for a generalization to densities of the form $e^{-G(x)}$, including normal distributions.

A.4 Proofs omitted from Section 9.5 (distributions on higher-order objects)

Lemma 9.1 (recap). *The Scott topology of $\llbracket \mathbf{int} \rightarrow \mathbf{real} \rrbracket$ has a subbase of open sets of the form $[n \in V] \stackrel{\text{def}}{=} \{f \in [\mathbb{Z}_{\perp} \rightarrow \mathbb{R}_{\perp}] \mid f(n) \in V\}$, where n ranges over \mathbb{Z}_{\perp} and V ranges over the open subsets of \mathbb{R}_{\perp} .*

The map j is a topological embedding of $\mathbb{R}^{\mathbb{N}}$ into $\llbracket \mathbf{int} \rightarrow \mathbf{real} \rrbracket$.

PROOF. \mathbb{Z}_{\perp} and \mathbb{R}_{\perp} are bc-domains, and therefore the Scott topology on $[\mathbb{Z}_{\perp} \rightarrow \mathbb{R}_{\perp}]$ coincides with the topology of pointwise convergence [32, Proposition 11.2]. The latter is the topology generated by the indicated subbasic open sets $[n \in V]$, by definition. This can also be deduced from Proposition II-4.6 of [27].

The inverse image $j^{-1}([n \in V])$ is equal to the subbasic open set $\pi_n^{-1}(U) \stackrel{\text{def}}{=} \{\vec{x} \in \mathbb{R}^{\mathbb{N}} \mid x_n \in U\}$ (where $U \stackrel{\text{def}}{=} i^{-1}(V)$) if $n \in \mathbb{N}$, is empty if $n \notin \mathbb{N}$ and $\perp \notin V$, and is the whole of $\mathbb{R}^{\mathbb{N}}$ otherwise. Therefore j is continuous.

Finally, every subbasic open set $\pi_n^{-1}(U)$ ($n \in \mathbb{N}$, U open in \mathbb{R}) is equal to $j^{-1}([n \in V])$, where V is any open subset of \mathbb{R}_{\perp} such that $i^{-1}(V) = U$ (which exists since i is an embedding). Hence j is a topological embedding. \square

Lemma 9.2 (recap). *The function $\text{bin}: \mathbb{R} \rightarrow \{0, 1\}^{\mathbb{N}}$ that maps every number in $[0, 1[$ to the string of bits in its binary expansion (namely, the i th bit in $\text{bin}(x)$ is $\lfloor 2^i \cdot x \rfloor$, for any $i \geq 1$), every negative number to the all zero string 0^ω , and every number larger than or equal to 1 to the all one string 1^ω is measurable. The image measure $\text{bin}[\lambda_1]$ is the unique measure v on $[0, 1]$ such that $v(\{s \in \{0, 1\}^{\mathbb{N}} \mid s_0 = b_0, \dots, s_{k-1} = b_{k-1}\}) = 1/2^k$ for all $k \in \mathbb{N}$ and elements $b_0, \dots, b_{k-1} \in \{0, 1\}$.*

PROOF. For every finite string $b \stackrel{\text{def}}{=} (b_0, b_1, \dots, b_{k-1})$ of bits, let U_b be the set of infinite strings of bits having b as a prefix. The sets U_b form a countable base of the topology on $\{0, 1\}^{\mathbb{N}}$, and $\text{bin}^{-1}(U_b)$ is the interval $[\sum_{i=0}^{k-1} b_i/2^{i+1}, \sum_{i=0}^{k-1} b_i/2^{i+1} + 1/2^k[$ if not all bits b_i are equal to 0 or to 1, $] - \infty, 1/2^k[$ if all the bits b_i are equal to 0, and to $[1 - 1/2^k, +\infty[$ otherwise. Every interval is in the Borel σ -algebra, and every open subset of $\{0, 1\}^{\mathbb{N}}$ is a countable union of sets U_s , so the inverse image of any open subset of $\{0, 1\}^{\mathbb{N}}$ by bin is Borel. It follows that bin is measurable. Then $\text{bin}[\lambda_1](U_b) = \lambda_1(\text{bin}^{-1}(U_b)) = \lambda(\text{bin}^{-1}(U_s) \cap [0, 1])$ is equal to $1/2^k$. Finally, the sets U_b form a π -system, whence the claim of uniqueness follows from the fact that any two probability measures that agree on a π -system generating the σ -algebra are equal [6, Theorem 3.3]. \square

Lemma 9.3 (recap). *The map $\text{num}: \{0, 1\}^{\mathbb{N}} \rightarrow \mathbb{R}$ that sends s to $\sum_{i \in \mathbb{N}} s_i/2^{i+1}$ is continuous, and the image measure $\text{num}[v]$ is equal to λ_1 .*

PROOF. Let U be any open subset of \mathbb{R} , and $s \in \text{num}^{-1}(U)$. For some $\epsilon > 0$, the interval $] \text{num}(s) - \epsilon, \text{num}(s) + \epsilon[$ is included in U . We pick $k \in \mathbb{N}$ so that $1/2^k < \epsilon$. Then the set of elements $t \in \{0, 1\}^{\mathbb{N}}$ such that $t_0 = s_0, \dots, t_k = s_k$ is an open neighborhood of s that is included in $\text{num}^{-1}(U)$, showing that $\text{num}^{-1}(U)$ is open. Therefore num is continuous.

It is easy to see that $\text{num}(\text{bin}(x)) = x$ for every $x \in [0, 1]$, so that $(\text{num} \circ \text{bin})^{-1}(E)$ and E have the same intersection with $[0, 1]$, for every subset E of \mathbb{R} . Hence, for every Borel measurable subset E of \mathbb{R} , and using Lemma 9.2, $\text{num}[v](E) = \text{num}[\text{bin}[\lambda_1]](E) = \lambda((\text{num} \circ \text{bin})^{-1}(E) \cap [0, 1]) = \lambda(E \cap [0, 1]) = \lambda_1(E)$. \square

Lemma 9.4 (recap). *The map $\text{split}: s \mapsto (s[0], \uparrow s)$ is a homeomorphism of $\{0, 1\}^{\mathbb{N}}$ onto $(\{0, 1\}^{\mathbb{N}})^2$. The image continuous valuation (resp., measure) $\text{split}[v]$ is the product valuation (resp., product measure) $v \otimes v$.*

PROOF. The first part is a direct consequence of the fact that the pairing map is bijective. For the second part, we reason on continuous valuations, since that will be slightly easier. For every pair of disjoint finite subsets I and J of \mathbb{N} , let $U_{I,J}$ be the set of elements s of $\{0, 1\}^{\mathbb{N}}$ such that $s_m = 0$ for every $m \in I$ and $s_n = 1$ for every $n \in J$. We see that those sets form a base of the topology on $\{0, 1\}^{\mathbb{N}}$, and that $v(U_{I,J}) = 1/2^{|I|+|J|}$, where $|_$ denotes cardinality, using the last part of Lemma 9.2. A base of the product topology on $(\{0, 1\}^{\mathbb{N}})^2$ is given by the sets $U_{I,J} \times U_{I',J'}$, and:

$$\begin{aligned} \text{split}[v](U_{I,J} \times U_{I',J'}) &= v(\text{split}^{-1}(U_{I,J} \times U_{I',J'})) \\ &= v(U_{I'',J''}), \end{aligned}$$

where $I'' \stackrel{\text{def}}{=} \{(0, k) \mid k \in I\} \cup \{(m', k) \mid m', k \in \mathbb{N}, \langle m', k \rangle \in I'\}$ and $J'' \stackrel{\text{def}}{=} \{(0, k) \mid k \in J\} \cup \{(n', k) \mid n', k \in \mathbb{N}, \langle n', k \rangle \in J'\}$. Hence $\text{split}[v](U_{I,J} \times U_{I',J'}) = 1/2^{|I''|+|J''|} = 1/2^{|I|+|I'|+|J|+|J'|}$. We also have:

$$(v \otimes v)(U_{I,J} \times U_{I',J'}) = v(U_{I,J}) \cdot v(U_{I',J'}) = 1/2^{|I|+|J|} \cdot 1/2^{|I'|+|J'|},$$

so $\text{split}[v]$ and $v \otimes v$ coincide on the basic open sets $U_{I,J}$. Using modularity, they must coincide on any finite disjoint union of basic open sets, and by Scott-continuity, they must coincide on any arbitrary disjoint union of basic open sets. (This is an argument we have sketched in the proof of Theorem 4.1 already.) Since any two basic open sets $U_{I,J}$ are either comparable or disjoint, every open subset of $\{0, 1\}^{\mathbb{N}}$ is such a disjoint union, and this concludes the proof. \square

Theorem 9.6 (recap). For every environment ρ , $\llbracket \text{rand_uniform_seq} \rrbracket \rho = j[\lambda_1^\infty]$.

PROOF. Let $\pi_N: \llbracket \mathbf{int} \rightarrow \mathbf{real} \rrbracket$ map f to $(f(0), f(1), \dots, f(N-1))$. We start by fixing N , and we compute the value of $\pi_N[\llbracket \text{rand_uniform_seq} \rrbracket \rho]$. This is the valuation that maps every open subset U of \mathbb{R}_\perp^N to the probability that a function f drawn at random according to $\llbracket \text{rand_uniform_seq} \rrbracket \rho$ produces a tuple $(f(0), f(1), \dots, f(N-1))$ that falls in U . Let us begin by noting that $\llbracket \text{rand_uniform_seq} \rrbracket \rho = \text{mux}[i[\lambda_1]]$.

Let $g: \mathbb{R} \rightarrow \mathbb{R}^{\mathbb{N}}$ map x to $(\text{num}(\text{bin}(x)[m]))_{m \in \mathbb{N}}$. It is easy to see that g is measurable. By definition, $g(x)_m = \text{mux}(i(x))(m)$ for every non-dyadic number x in $[0, 1]$, namely for every $x \in \mathbb{R} \setminus E_0$ where the set E_0 was introduced in Remark 9.1, and for every $m \in \mathbb{N}$.

We claim that $j[g[\lambda_1]] = \text{mux}[i[\lambda_1]]$. This needs a reminder and a comment. We recall that j is the canonical embedding of $\mathbb{R}^{\mathbb{N}}$ into $\llbracket \mathbf{int} \rightarrow \mathbf{real} \rrbracket$; and we have silently promoted λ_1 from a continuous valuation to a measure on \mathbb{R} , therefore working with image *measures* rather than image *valuations*.

The claim is proved as follows. For every Borel measurable subset E of $\llbracket \mathbf{int} \rightarrow \mathbf{real} \rrbracket$, $j[g[\lambda_1]](E) = \lambda_1((j \circ g)^{-1}(E))$, while $\text{mux}[i[\lambda_1]] = \lambda_1((\text{mux} \circ i)^{-1}(E))$. The symmetric difference of $(j \circ g)^{-1}(E)$ and of $(\text{mux} \circ i)^{-1}(E)$ is included in E_0 , which has λ_1 -measure zero, so $j[g[\lambda_1]](E) = \lambda_1((j \circ g)^{-1}(E))$.

Let us write $i^{(N)}$ for the N -fold product $i \times \dots \times i$, and similarly with $\text{num}^{(N)}$. For every $x \in \mathbb{R}$, $\pi_N(\text{mux}(i(x))) = (i(g(x)_0), \dots, i(g(x)_{N-1})) = (i(\text{num}(\text{bin}(x)[0])), \dots, i(\text{num}(\text{bin}(x)[N-1]))) = (i^{(N)} \circ \text{num}^{(N)} \circ \omega_N \circ \text{bin})(x)$. It follows:

$$\begin{aligned}
 \pi_N[\text{mux}[i[\lambda_1]]] &= i^{(N)}[\text{num}^{(N)}[\omega_N[\text{bin}[\lambda_1]]]] \\
 &= i^{(N)}[\text{num}^{(N)}[\omega_N[v]]] && \text{by Lemma 9.2} \\
 &= i^{(N)}[\text{num}^{(N)}[v \otimes \dots \otimes v]] && \text{by Corollary 9.5, item 3} \\
 &= i[\text{num}[v]] \otimes \dots \otimes i[\text{num}[v]] \\
 &= i[\lambda_1] \otimes \dots \otimes i[\lambda_1] && \text{by Lemma 9.3.}
 \end{aligned}$$

This is a statement about measures. Restricting to open sets, we obtain the following statement about continuous valuations:

Fact A.1. $\pi_N[\llbracket \text{rand_uniform_seq} \rrbracket \rho]$ is the N -fold valuation product of $i[\lambda_1]$.

We use this to prove that $\llbracket \text{rand_uniform_seq} \rrbracket \rho j[\lambda_1^\infty]$, qua continuous valuations. It suffices to show that the two sides of the equation coincide on basic open subsets of $\llbracket \mathbf{int} \rightarrow \mathbf{real} \rrbracket$: as in the proof of Theorem 4.1 or of Lemma 9.4, modularity and Scott-continuity will imply they they also coincide on every open set at all.

As basic open subsets, Lemma 9.1 suggests that we take the finite intersections of sets of the form $[n \in V]$. Without loss of generality, let $U \stackrel{\text{def}}{=} [0 \in V_0] \cap [1 \in V_1] \cap \dots \cap [N-1 \in V_{N-1}]$. (Any set of the form $[n \in V_n]$ where V_n is the whose of \mathbb{R}_\perp can be inserted at will, since they will not modify the intersection.) We note that $U = \pi_N^{-1}(V_0 \times V_1 \times \dots \times V_{N-1})$, so:

$$\begin{aligned}
 \llbracket \text{rand_uniform_seq} \rrbracket \rho(U) &= \pi_N[\llbracket \text{rand_uniform_seq} \rrbracket \rho](V_0 \times V_1 \times \dots \times V_{N-1}) \\
 &= \prod_{n=0}^{N-1} i[\lambda_1](V_n) && \text{by Fact A.1.}
 \end{aligned}$$

Now, since $j(\vec{x})(n) = i(x_n)$ for all $\vec{x} \in [0, 1]^{\mathbb{N}}$ and $n \in \mathbb{N}$, $j^{-1}(U) = \pi_N^{-1}(i^{-1}(V_0) \times i^{-1}(V_1) \times \cdots \times i^{-1}(V_{N-1}))$, so:

$$\begin{aligned} j[\lambda_1^\infty](U) &= \pi_N[\lambda_1^\infty](i^{-1}(V_0) \times i^{-1}(V_1) \times \cdots \times i^{-1}(V_{N-1})) \\ &= \prod_{n=0}^{N-1} \lambda_1(i^{-1}(V_n)), \end{aligned}$$

by definition of λ_1^∞ , and this concludes the proof. \square

A.5 More distributions on higher-order objects

At this point, let us be less formal. The construction of λ_1^∞ through `rand_uniform_seq` opens up several avenues, which we mostly leave as programming exercises to the reader.

For one, we can now implement the Box-Muller algorithm (see Section 9.2) by drawing just *one* random real. We remember that the Box-Muller algorithm uses rejection sampling, which may require us to draw arbitrarily many independent, random, λ_1 -distributed real numbers. Instead, we can just call `rand_uniform_seq` once, obtaining a random, λ_1^∞ -distributed function f (modulo the embedding j); each time we need another random real number, we simply read the next value $f(n)$, for larger and larger values of n .

Going further, instead of calling `rand_uniform_seq`, we may call `mux` on a random λ_1 -distributed real number given as argument. We obtain an ISPCF term `box_muller_transform` of type $\mathbf{real} \rightarrow \mathbf{real} \times \mathbf{real}$ (not $\mathbf{real} \rightarrow D(\mathbf{real} \times \mathbf{real})$), since that term no longer draws anything at random by itself) that maps a random λ_1 -distributed real number to a pair of two independent, random $\mathcal{N}(0, 1)$ -distributed real numbers.

```

box_muller_transform  $\stackrel{\text{def}}{=} \lambda \text{seed}_{\mathbf{real}}.$ 
  let randnumsint→real = mux seed in
  let xreal = randnums 0 in
  let yreal = randnums 1 in
  let creal = cos(2.0 × π × y) in
  let sreal = sin(2.0 × π × y) in
  let mreal = sqrt(-2.0 × log x) in
  ⟨m × c, m × s⟩.

```

We let the reader check that the semantics of:

```
do seed ← sample[λ1]; ret(box_muller_transform seed)
```

is $i[\mathcal{N}(0, 1) \otimes \mathcal{N}(0, 1)]$, just like `box_muller`.

The tricks described above suggest the following scheme for implementing the infinite product $\mathcal{N}(0, 1)^\infty$. We call `rand_uniform_seq` in order to obtain an infinite, independent sequence of λ_1 -distributed real numbers. We then apply `box_muller_transform` to each. Explicitly, we form the ISPCF term:

```
do fint→real ← rand_uniform_seq; ret λnint. box_muller_transform(f n),
```

of type $D(\mathbf{int} \rightarrow \mathbf{real})$.

Again, this can be made into a transformer

`box_muller_sequence_transform: real → D(int → real)`

that maps any random λ_1 -distributed real number to a random $\mathcal{N}(0, 1)^\infty$ -distributed infinite sequence of real numbers. This can also be iterated. We let the reader explore around this idea. For example, given two maps $mean, sigma: \mathbf{int} \rightarrow \mathbf{real}$, build a term that maps any random λ_1 -distributed real number to a random infinite sequence of independent real numbers, whose n th entry follows a normal distribution with mean $mean(n)$ and with standard deviation $sigma(n)$.

One can implement infinite independent sequences of random real numbers following various other distributions in the same way. For example, do this for infinite products of exponential distributions.

This machinery can also be used to implement non-trivial distributions over distributions. A typical, and useful, application is the *Dirichlet process* [22], which is widely used as a source of Bayesian priors in the field of nonparametric estimation problems. Given a base distribution H over some space X , and a parameter $\alpha \in]0, 1[$, the Dirichlet process $DP(H)$ is a distribution over the space of probability distributions over X , which one may describe as follows. We draw an infinite sequences of elements $x_n, n \in \mathbb{N}$, from X . At step n , with probability $\alpha/(\alpha + n)$, we draw x_n from X , independently from all previous values, using the distribution H ; otherwise, we draw x_n at random among the previous values, each having the same probability, in other words with respect to the distribution $\frac{1}{n} \sum_{i=0}^{n-1} \delta_{x_i}$. (This is undefined if $n = 0$, but then, if $n = 0$, that second case happens with zero probability.) This process yields a random infinite sequence, which we can use directly, or which we can convert to a distribution, the *directing measure*, mapping every measurable subset E of X to the limit of $(\frac{1}{n} \sum_{i=0}^{n-1} \delta_{x_i})(E)$ as n tends to $+\infty$, which exists H -almost surely by de Finetti's theorem [47, Chapter 1.1], based on the fact that the sequence $(x_n)_{n \in \mathbb{N}}$ is [what is known as an](#) exchangeable sequence.

In principle, one can compute the directing measure from any exchangeable sequence, as shown by Daniel Roy [65, Chapter IV]. In the special case of the Dirichlet process, there is a popular, and more efficient, implementation of $DP(H)$, using the *stick-breaking algorithm*, see Figure 2 of [29] for example. (We also refer to that paper for an application of the Dirichlet process to an interesting, stochastic form of memoization.) While the latter relies on memoizing facilities (hence on global state, possibly hidden), one can implement it in ISPCF as follows, given H^∞ (not $H!$) as argument.

We assume that X is the denotation $\llbracket \tau \rrbracket$ of some type τ , and that the variable $proc_{D(\mathbf{int} \rightarrow \tau)}$ holds a representation of H^∞ . We draw a random function $atoms: \mathbf{int} \rightarrow \tau$ with respect to H^∞ , representing some infinite sequence $(\theta_n)_{n \in \mathbb{N}}$ of elements of X ; this uses `do` and `proc`. We also draw a random function $sticks: \mathbf{int} \rightarrow \mathbf{real}$ with respect to the distribution $\text{Beta}(1, \alpha)^\infty$ (the infinite product of the β distribution with parameters 1 and α), representing a random sequence $(\beta'_n)_{n \in \mathbb{N}}$. We define a new random (but no longer independent) sequence $(\beta_n)_{n \in \mathbb{N}}$, where $\beta_n \stackrel{\text{def}}{=} \beta'_n \cdot \prod_{i=0}^{n-1} (1 - \beta'_i)$. and we return the (random) distribution $\sum_{n \in \mathbb{N}} \beta_n \delta_{\theta_n}$.

While that would be possible in ISPCF, we do not compute any β_n explicitly. Instead, the final distribution is obtained by the following informal procedure: with probability β'_0 , return θ_0 ; else, with probability β'_1 , return θ_1 ; else, with probability β'_2 , return θ_2 , and so on. This terminates with probability 1.

This can be implemented in ISPCF by drawing a third function $rand: \mathbf{int} \rightarrow \mathbf{real}$ at random with respect to λ_1^∞ , then returning $atoms$ ($pick_a_stick\ sticks\ rand\ 0$), where:

$$\begin{aligned} pick_a_stick &\stackrel{\text{def}}{=} \lambda sticks_{\mathbf{int} \rightarrow \mathbf{real}}. \lambda rand_{\mathbf{int} \rightarrow \mathbf{real}}. \mathbf{rec}(\lambda pick_{\mathbf{int} \rightarrow \mathbf{int}}. \\ &\quad \lambda j_{\mathbf{int}}. \\ &\quad \mathbf{if}(\underline{\text{pos}}(sticks\ j - rand\ j)) \\ &\quad \quad j \\ &\quad (pick\ (j + 1))). \end{aligned}$$

We have taken the same procedure names as in [29, Figure 2], in the hope that this will ease a comparison with the Church implementation of the Dirichlet process given there.

B OTHER OMITTED PROOFS

B.1 Proof of Proposition 10.8

Proposition 10.8 (recap). *Let Σ consist of strictly observable first-order constants. The map $Next$ is Scott-continuous from Γ to \mathbf{VT} .*

PROOF. Let L be any generalized ISPCF term, \mathcal{V} be any Scott-open subset of \mathbf{VT} , and let us assume that $Next(L) \in \mathcal{V}$. We will show that there is a Scott-open neighborhood U of L such that $U \subseteq Next^{-1}(\mathcal{V})$. U will consist of terms of the same shape P as L .

If L is of the form $E[\mathbf{rec}\ M]$, then $Next(L) = \delta_R$ where $R \stackrel{\text{def}}{=} E[M(\mathbf{rec}\ M)]$ is computed as follows: writing L as $P\theta$ for some unique configuration (P, θ) , R is obtained as $Q\theta'$ for some unique configuration (Q, θ') , where Q is obtained from P alone (independently of θ), and $\theta' = \theta \circ \sigma$ for some function $\sigma: \text{fv}(Q) \rightarrow \text{fv}(P)$ that is determined from P alone, again. For example, if $L = \underline{1.0} + \mathbf{rec}(\lambda x.x \times \underline{2.0})(\underline{3.0})$, then $P = x_1 + \mathbf{rec}(\lambda x.x \times x_2)(x_3)$, $Q = x_1 + (\lambda x.x \times x_2)(\mathbf{rec}(\lambda x.x \times x_3))(x_4)$, and σ maps x_1 to x_1 , x_2 and x_3 to x_2 , and x_4 to x_3 .

Since δ_R is in \mathcal{V} , R is in $V \stackrel{\text{def}}{=} \eta_\Gamma^{-1}(\mathcal{V})$, and θ' is in the open subset $\{\theta' \in \mathbb{R}^{\text{fv}(Q)} \mid Q\theta' \in V\}$. The map $S: \theta \mapsto \theta \circ \sigma$ is clearly Scott-continuous. It follows that $U \stackrel{\text{def}}{=} \{P\theta \mid \theta \in S^{-1}(\{\theta' \in \mathbb{R}^{\text{fv}(Q)} \mid Q\theta' \in V\})\}$ is Scott-open in Γ . Moreover, by definition, for every generalized ISPCF term $P\theta$ in U , $Next(P)$ is in \mathcal{V} .

We reason similarly for all other rules except (5), (6), and (7). The case of (7) is not that different. In the final step, U is equal to the intersection of $\{P\theta \mid \theta \in S^{-1}(\{\theta' \in \mathbb{R}^{\text{fv}(Q)} \mid Q\theta' \in V\})\}$ with the set U' of generalized ISPCF terms of shape P on which rule (7) applies; then we will be able to conclude as above, once we prove that U' is open. In order to do so, and since we are in the case of (7), P is of the form $E[\underline{f}\ M_1 \cdots M_k]$, where each M_i is either a constant (if its type is not \mathbf{real}) or a template variable. Let x_{j+1}, \dots, x_{j+p} be the template variables occurring among the terms M_i , listed from left to right, and say that $x_{j+1} = M_{i_1}, \dots, x_{j+p} = M_{i_p}$, where $1 \leq i_1 < \dots < i_p \leq k$. Let also M_i be the constant \underline{a}_i for each i not among i_1, \dots, i_p . Then $U' = \{P\theta \mid \theta \in \mathbb{R}^{\text{fv}(P)}, f(a_1) \cdots (a_{i_1-1})(\theta(x_{j+1}))(a_{i_1+1}) \cdots (a_{i_2-1})(\theta(x_{j+2}))(a_{i_2+1}) \cdots (a_{i_p-1})(\theta(x_{k+p}))(a_{i_p+1}) \cdots (a_k) \in \text{Obs}_\beta\}$, where β is the (observable) type of $f\ M_1 \cdots M_k$, and Obs_β is the subset of observable elements of $\llbracket \beta \rrbracket$. It is easy to see that Obs_β is Scott-open. Since f is Scott-continuous, U' is open.

In the case of (6), $L = E^0[\mathbf{score}\ \underline{a}]$. We write L as $P\theta$ for some unique configuration (P, θ) , once again. Then $Next(L) = |\underline{a}|. \delta_{E[\mathbf{ret}\ *]}$ where $E^0[\mathbf{ret}\ *]$ can be written as $Q\theta'$ for some configuration (Q, θ') . Once again, Q is determined as a function of P alone. In fact, if \underline{a} is the m th occurrence of a constant of type \mathbf{real} in L (so that $\theta(x_m) = \underline{a}$;

recall that x_m is the m th template variable), and $\text{fv}(P) = \{x_1, \dots, x_n\}$ with $n \geq m$, then $\text{fv}(Q) = \{x_1, \dots, x_{n-1}\}$, and $\theta' = \theta \circ \sigma$, where σ maps x_1 to x_1, \dots, x_{m-1} to x_{m-1} , and x_m to x_{m+1}, \dots, x_{n-1} to x_n . The map $\mathbf{a} \mapsto |\mathbf{a}|$ is Scott-continuous from \mathbb{R} to $\overline{\mathbb{R}}_+$. Product is a Scott-continuous map on $\overline{\mathbb{R}}_+$, from which we obtain easily that $(a, v) \mapsto av$ is Scott-continuous from $\overline{\mathbb{R}}_+ \times \mathbb{V}\Gamma$ to $\mathbb{V}\Gamma$. Using that η_Γ is also Scott-continuous, the map $(\mathbf{a}, \theta') \in \mathbb{R} \times \mathbb{R}^{\text{fv}(Q)} \mapsto |\mathbf{a}| \cdot \delta_{Q\theta'}$ is Scott-continuous, and therefore $S: \theta \in \mathbb{R}^{\text{fv}(P)} \mapsto |\theta(x_m)| \cdot \delta_{Q(\theta \circ \sigma)}$ is also Scott-continuous. We then define U as $\{P\theta \mid \theta \in S^{-1}(\mathcal{V})\}$. This is open, contains L , and its image by Next is included in \mathcal{V} by definition.

In the case of (5), finally, $L = E^0[\mathbf{sample}[\mu]]$. This time, $L = P\theta$ for some unique configuration (P, θ) . For every $a \in \mathbb{R}$, $f_{E^0}(a) = E^0[\mathbf{ret} \underline{a}]$ can be written as $Q\theta'(a)$, where $(Q, \theta'(a))$ is a configuration such that Q is obtained from P alone, as in previous cases. Additionally, there is a number m (the number of the template variable that is replaced by \underline{a} in Q), the template variables of P are x_1, \dots, x_n with $n \geq m$, the template variables of Q are x_1, \dots, x_{n+1} , and θ' is obtained as $S(\theta, i(a))$, where the map S is defined by: for all $\theta \in \mathbb{R}^{\text{fv}(P)}$ and $\mathbf{a} \in \mathbb{R}$, $S(\theta, \mathbf{a})$ maps x_m to \underline{a} , every x_i with $i < m$ to $\theta(x_i)$, and every x_i with $i > m$ to $\theta(x_{i-1})$. S is clearly Scott-continuous. In particular, $f_{E^0} = Q S(\theta, i(_))$ is lower semicontinuous, so $f_{E^0}[\mu]$ makes sense. Moreover, the map $\theta \mapsto f_{E^0}(a) = Q S(\theta, i(a))$ is also Scott-continuous for every $a \in \mathbb{R}$, so the map that sends every θ to $f_{E^0}[\mu] = \lambda V \in \mathcal{O}\Gamma. \mu \{a \in \mathbb{R} \mid Q S(\theta, i(a)) \in V\}$ is also Scott-continuous. Therefore $U \stackrel{\text{def}}{=} \{P\theta \in \mathbb{R}^{\text{fv}(P)} \mid f_{E^0}[\mu] \in \mathcal{V}\}$ is open. By construction, L is in U , and every element of U maps to an element of \mathcal{V} by Next . \square

B.2 Soundness (Section 10.1)

Lemma 10.11 (recap). *Let Σ consist of strictly observable first-order constants. The weak normal forms are exactly the lazy values.*

PROOF. It is clear that every lazy value is in weak normal form. Conversely, we show that every weak normal form M is a lazy value V^0 , by induction on the size of M .

We start with the case where M is a constant \underline{f} . If $\alpha(\underline{f}) = 0$, then M is a zero-ary constant of observable type. By the last condition of Definition 10.5, it is of basic type, and therefore it is a lazy value. If $\alpha(\underline{f}) \neq 0$, then it is a lazy value of the form $\underline{f} V_1 \cdots V_i$, with $i < \alpha(\underline{f})$ (namely with $i = 0$).

If M is of the form $\mathbf{score} N$, then N is weakly normal, since $\mathbf{score}[]$ is a weak evaluation context and M is weakly normal. We apply the induction hypothesis to N , and we realize that because of typing, N must be of the form \underline{a} for some zero-ary constant \underline{a} of type \mathbf{real} . However, $M = \mathbf{score} \underline{a}$ is not weakly normal, so M cannot be of the form $\mathbf{score} N$ after all.

M cannot be of the form $\mathbf{rec} N$ or $\mathbf{sample}[\mu]$, which are not weakly normal. Lambda-abstractions, and terms of the form $\mathbf{ret} N$, $\langle N, P \rangle$, $\iota_1 N$, or $\iota_2 N$ are all lazy values.

If M is of the form $\pi_1 N$, then N is weakly normal, so by induction hypothesis and typing, N is a pair, which would contradict the normality of M ; similarly if $M = \pi_2 N$. If M is of the form $\mathbf{case} NPQ$, then N is weakly normal, so by induction hypothesis and typing it is of the form $\iota_1 N_1$ or $\iota_2 N_2$, and that would contradict the normality of M as well.

If M is of the form $\mathbf{do} x \leftarrow N; P$, then by induction hypothesis and typing N is of the form $\mathbf{ret} P$, which would contradict the fact that M is weakly normal.

If M is an application NP , then by induction hypothesis and typing, either N is a λ -abstraction (which is impossible since M is weakly normal), or N is of the form $\underline{f} M_1 \cdots M_i$ with $i < \alpha(\underline{f})$ and where M_1, \dots, M_i are weakly normal. Then $M = \underline{f} M_1 \cdots M_i P$, where M_1, \dots, M_i and P are weakly normal, hence are lazy values. By the first condition of

Definition 10.5, they are lazy values of basic types, hence are constants of arity 0. If $i + 1 = \alpha(\underline{f})$, either rule (7) would apply or M would be blocked. Hence $i + 1 < \alpha(\underline{f})$, showing that M is a lazy value. \square

Lemma 10.12 (recap). *Let Σ consist of strictly observable first-order constants. For every type τ , $Norm_\tau$ is exactly the set of values of type τ .*

PROOF. It is clear that every value is in normal form. Conversely, we argue that every element M of $Norm_\tau$ is a value, by induction on M . Every normal form is weakly normal, since every weak evaluation context E^0 is a context E ; so M is a lazy value. In order to show that it is a value, we only need to show that if it is of the form $\mathbf{ret} N$, $\langle N, P \rangle$, $\iota_1 N$ or $\iota_2 N$, then N and P are values. If $M = \mathbf{ret} N$, this is because $\mathbf{ret}[\]$ is an evaluation context, so that N is a value by induction hypothesis. Similarly when $M = \iota_1 N$ or $M = \iota_2 N$. When $M = \langle N, P \rangle$, since $\langle [\], P \rangle$ is an evaluation context, by induction hypothesis N is a value V . Since $\langle V, [\] \rangle$ is a context, P is a value. \square

Lemma 10.13 (recap). *Let X be a topological space, and let $Norm$ and \overline{Norm} be two complementary open subsets of X . Let $\kappa: X \rightarrow \mathbf{V}X$ be a continuous map, and let us assume that for every $x \in Norm$, $\kappa(x)$ is the zero valuation. We define:*

- $\kappa^{\leq 0}$ as mapping every $x \in Norm$ to δ_x , and every $x \in \overline{Norm}$ to the zero valuation;
- $\kappa^{\leq 1} \stackrel{\text{def}}{=} \kappa + \kappa^{\leq 0}$;
- $\kappa^{\leq n+1} \stackrel{\text{def}}{=} (\kappa^{\leq n})^\dagger \circ \kappa^{\leq 1}$, for every $n \geq 1$.

Then:

- (1) the maps $\kappa^{\leq n}$ are continuous from X to $\mathbf{V}X$;
- (2) for every $x \in X$, the family $(\kappa^{\leq n}(x)|_{Norm})_{n \geq 1}$ is monotonically increasing;
- (3) the formula $\kappa^*(x) \stackrel{\text{def}}{=} \sup_{n \geq 1} \kappa^{\leq n}(x)|_{Norm}$ defines a continuous map κ^* from X to $\mathbf{V}X$.

PROOF. 1. Since X is homeomorphic to the topological coproduct $\overline{Norm} + Norm$, the continuity of $\kappa^{\leq 0}$ reduces to that the unit η_{Norm} and of the constant zero map. Then $\kappa^{\leq 1}$ is continuous since $+$ is Scott-continuous on $\overline{\mathbb{R}}_+$, and by induction on $n \geq 1$, $\kappa^{\leq n+1}$ is continuous as a composition of continuous maps.

2. We show that $\kappa^{\leq n}(x)(U) \leq \kappa^{\leq n+1}(x)(U)$ for every $x \in X$ and every open subset U of $Norm$ by induction on $n \in \mathbb{N}$.

In the base case $n = 0$, this is obvious. Otherwise, $n \geq 1$, and $\kappa^{\leq n+1}(x)(U) = \int_{y \in X} \kappa^{\leq n}(y)(U) d\kappa^{\leq 1}(x)$. For every $y \in X$, $\kappa^{\leq n}(y)(U) \geq \kappa^{\leq n-1}(y)(U)$ by induction hypothesis, so $\kappa^{\leq n+1}(x)(U) \geq \int_{y \in X} \kappa^{\leq n-1}(y)(U) d\kappa^{\leq 1}(x) = \kappa^{\leq n}(x)(U)$.

3. $\mathbf{V}X$ is a dcpo, and suprema in the dcpo of continuous maps from X to $\mathbf{V}X$ are computed pointwise. \square

Lemma 10.14 (recap). *Let Σ consist of strictly observable first-order constants. For every generalized ISPCF term M , for every $n \in \mathbb{N}$, we have:*

- (1) if $M \in Norm$, then $Next^{\leq n}(M) = \delta_M$;

PROOF. By induction on n . If $n = 0$, then $Next^{\leq 0}(M) = \delta_M$ since $M \in Norm$. If $n = 1$, then $Next(M) = 0$, so $Next^{\leq 1}(M) = Next(M) + Next^{\leq 0}(M) = \delta_M$. If $n \geq 2$, then $Next^{\leq n}(M) = (Next^{\leq n-1})^\dagger(Next^{\leq 1}(M)) = (Next^{\leq n-1})^\dagger(\delta_M) = Next^{\leq n-1}(M) = \delta_M$, by induction hypothesis. \square

- (2) if $M = E^0[\mathbf{sample}[\mu]]$, and $n \geq 1$, then for every $U \in O\Gamma$, $Next^{\leq n}(M)(U) = \int_{a \in \mathbb{R}} Next^{\leq n-1}(E^0[\mathbf{ret} a])(U) d\mu$;

PROOF. Since $M = E^0[\mathbf{sample}[\mu]] \in \overline{Norm}$, $Next^{\leq 1}(M) = Next(M)$ is equal to $f_{E^0}[\mu]$, where $f_{E^0}(a) \stackrel{\text{def}}{=} E^0[\mathbf{ret} a]$ for every $a \in \mathbb{R}$. Therefore $Next^{\leq n}(M)(U) = (Next^{\leq n-1})^\dagger(f_{E^0}[\mu])(U) = \int_{N \in \Gamma} Next^{\leq n-1}(N)(U) df_{E^0}[\mu]$.

By the change-of-variables formula, this is equal to $\int_{a \in \mathbb{R}} \text{Next}^{\leq n-1}(f_{E^0}(a))(U) d\mu = \int_{a \in \mathbb{R}} \text{Next}^{\leq n-1}(E^0[\mathbf{ret} \underline{a}])(U) d\mu$. \square

(3) if $M = E^0[\mathbf{score} \underline{a}]$ and $n \geq 1$, then $\text{Next}^{\leq n}(M) = |\mathbf{a}|.\text{Next}^{\leq n-1}(E^0[\mathbf{ret} *]);$

PROOF. Since $M = E^0[\mathbf{score} \underline{a}] \in \overline{\text{Norm}}$, $\text{Next}^{\leq 1}(M) = \text{Next}(M) = |\mathbf{a}|.\delta_{E^0[\mathbf{ret} *]}$. Then, for every $U \in \mathcal{O}\Gamma$, $\text{Next}^{\leq n}(M)(U) = (\text{Next}^{\leq n-1})^\dagger(|\mathbf{a}|.\delta_{E^0[\mathbf{ret} *]})(U) = \int_{N \in \Gamma} \text{Next}^{\leq n-1}(N)(U) d|\mathbf{a}|.\delta_{E^0[\mathbf{ret} *]}$, and this is equal to $|\mathbf{a}|.\text{Next}^{\leq n-1}(E^0[\mathbf{ret} *])(U)$. \square

(4) if $M = E[f \underline{a_1} \cdots \underline{a_k}]$ and $n \geq 1$, then $\text{Next}^{\leq n}(M)$ is equal to $\text{Next}^{\leq n-1}(E[f \underline{a_1} \cdots \underline{a_k}])$ if $f(a_1) \cdots (a_k)$ is observable, to the zero valuation otherwise;

PROOF. Let $M = E[f \underline{a_1} \cdots \underline{a_k}]$. If $f(a_1) \cdots (a_k)$ is not observable, then M is blocked, so $\text{Next}^{\leq 1}(M) = 0$. For every $U \in \mathcal{O}\Gamma$, $\text{Next}^{\leq n}(M)(U) = (\text{Next}^{\leq n-1})^\dagger(0) = \int_{N \in \Gamma} \text{Next}^{\leq n-1}(N)(U) d0 = 0$, so $\text{Next}^{\leq n}(M) = 0$.

If $f(a_1) \cdots (a_k)$ is observable, then $\text{Next}^{\leq 1}(M) = \text{Next}(M) = \delta_{E[f(a_1) \cdots (a_k)]}$. It follows that $\text{Next}^{\leq n}(M) = (\text{Next}^{\leq n-1})^\dagger(\eta_\Gamma(E[f(a_1) \cdots (a_k)])) = \text{Next}^{\leq n-1}(E[f(a_1) \cdots (a_k)])$. \square

(5) for every instance of a rule $L \rightarrow R$ of Figure 3 except (5), (6) and (7), if $n \geq 1$ then $\text{Next}^{\leq n}(L) = \text{Next}^{\leq n-1}(R)$.

PROOF. This is similar to the latter case. \square

LEMMA B.1. *The only weak evaluation contexts E^0 whose type is of the form $D\sigma \vdash \tau$ are those of the form:*

$$\mathbf{do} x_n \leftarrow (\mathbf{do} x_{n-1} \leftarrow \cdots (\mathbf{do} x_1 \leftarrow []; N_1); \cdots; N_{n-1}); N_n,$$

and then τ is a distribution type $D\tau'$.

PROOF. By induction on the size of E^0 . This is obvious if $E^0 = []$, otherwise we see that E^0 must be of the form $E'^0[\mathbf{do} x \leftarrow []; N]$, by inspection of the evaluation context formation rules, using typing to rule out all the other possibilities. \square

The following linearity property is crucial for soundness. The importance of such linearity properties have already been made in [33, Proposition 5.1], where probabilistic choice was discrete, and in [17, Lemma 7.6]. Given any evaluation context E , we write $\llbracket E \rrbracket$ for $\llbracket \lambda x. E[x] \rrbracket$.

LEMMA B.2. *Assume that all constants in Σ are first-order constants. For every weak evaluation context E^0 of type $D\sigma \vdash D\tau$, $\llbracket E^0 \rrbracket$ is linear:*

- (1) for all $\mu, \mu' \in \mathbf{V}[\sigma]$, for every $a \in \mathbb{R}_+$, $\llbracket E^0 \rrbracket(a.\mu) = a \llbracket E^0 \rrbracket(\mu)$ and $\llbracket E^0 \rrbracket(\mu + \mu') = \llbracket E^0 \rrbracket(\mu) + \llbracket E^0 \rrbracket(\mu')$;
- (2) for every dcpo X , for every Scott-continuous map $f: X \rightarrow \mathbf{V}[\sigma]$, $\llbracket E^0 \rrbracket \circ f^\dagger = (\llbracket E^0 \rrbracket \circ f)^\dagger$.

PROOF. By Lemma B.1, E^0 is of the form $\mathbf{do} x_n \leftarrow (\mathbf{do} x_{n-1} \leftarrow \cdots (\mathbf{do} x_1 \leftarrow []; N_1); \cdots; N_{n-1}); N_n$. We prove both claims by induction on n , the length of E^0 .

1. If $n = 0$, this is obvious. Otherwise, we can write E^0 as $\mathbf{do} x_n \leftarrow E'^0; N_n$ where E'^0 has length $n - 1$. We note that, for every continuous map $f: X \rightarrow \mathbf{V}Y$, f^\dagger is linear, in the sense that $f^\dagger(a.\mu) = a.f^\dagger(\mu)$ and $f^\dagger(\mu + \mu') = f^\dagger(\mu) + f^\dagger(\mu')$. This is clear from the definition of f^\dagger , see formula (1). Then:

$$\begin{aligned} \llbracket E^0 \rrbracket(a.\mu) &= (\llbracket \lambda x_n. N_n \rrbracket)^\dagger(\llbracket E'^0 \rrbracket(a.\mu)) \\ &= (\llbracket \lambda x_n. N_n \rrbracket)^\dagger(a. \llbracket E'^0 \rrbracket(\mu)) && \text{by induction hypothesis} \\ &= a. (\llbracket \lambda x_n. N_n \rrbracket)^\dagger(\llbracket E'^0 \rrbracket(\mu)) = a. \llbracket E^0 \rrbracket(\mu) \end{aligned}$$

and similarly for $\llbracket E \rrbracket (\mu + \mu')$.

2. Again, this is clear when $n = 0$. When $n \geq 1$, we have:

$$\begin{aligned}
\llbracket E^0 \rrbracket \circ f^\dagger &= (\llbracket \lambda x_n. N_n \rrbracket)^\dagger \circ \llbracket E'^0 \rrbracket \circ f^\dagger \\
&= (\llbracket \lambda x_n. N_n \rrbracket)^\dagger \circ \left(\llbracket E'^0 \rrbracket \circ f \right)^\dagger && \text{by induction hypothesis} \\
&= \left((\llbracket \lambda x_n. N_n \rrbracket)^\dagger \circ \llbracket E'^0 \rrbracket \circ f \right)^\dagger && \text{by the equation } (g^\dagger \circ f)^\dagger = g^\dagger \circ f^\dagger \\
&= (\llbracket E^0 \rrbracket \circ f)^\dagger.
\end{aligned}$$

□

We can now establish soundness. **Proposition 10.17 (recap).** *Let Σ consist of strictly observable first-order constants. For every generalized ISPCF term $M : D\tau$, where τ is any type, for every open subset U of $\llbracket \tau \rrbracket$, $\llbracket M \rrbracket (U) \geq \text{Next}^*(M)(\overline{U})$.*

PROOF. We show that $\llbracket M \rrbracket \rho(U) \geq \text{Next}^{\leq n}(M)(\overline{U})$ for every $n \in \mathbb{N}$. If $M \in \text{Norm}$, then $\text{Next}^{\leq n}(M) = \delta_M$ by Lemma 10.14, item 1. By Lemma 10.12 (and typing), $M = \mathbf{ret} V$ for some value V of type τ . Hence $\llbracket M \rrbracket (U) = \delta_{\llbracket V \rrbracket}(U)$, which is equal to 1 if $\llbracket V \rrbracket \in U$, and to 0 otherwise. Since $\llbracket V \rrbracket \in U$ if and only if $V \in \overline{U}$ if and only if $\mathbf{ret} V \in \overline{U}$, $\llbracket M \rrbracket (U) = \text{Next}^{\leq n}(M)(\overline{U})$.

Henceforth, we assume that M is not normal, and we prove the claim by induction on n . If $n = 0$, then $\text{Next}^{\leq 0}(M) = 0$, so the claim is clear. Let therefore $n \geq 1$, and let us look at the possible shapes of M .

If $M = E^0[\mathbf{sample}[\mu]]$, then:

$$\begin{aligned}
\text{Next}^{\leq n}(M)(\overline{U}) &= \int_{a \in \mathbb{R}} \text{Next}^{\leq n-1}(E^0[\mathbf{ret} \underline{a}])(\overline{U}) d\mu && \text{by Lemma 10.14, item 2} \\
&\leq \int_{a \in \mathbb{R}} \llbracket E^0[\mathbf{ret} \underline{a}] \rrbracket (U) d\mu && \text{by induction hypothesis} \\
&= \int_{a \in \mathbb{I}\mathbb{R}_\perp} \llbracket E^0 \rrbracket (\eta_{\llbracket \mathbf{real} \rrbracket}(\underline{a}))(U) di[\mu] \\
&\quad \text{by the change-of-variables formula} \\
&= (\llbracket E^0 \rrbracket \circ \eta_{\llbracket \mathbf{real} \rrbracket})^\dagger(i[\mu])(U) && \text{by definition of } \dagger \\
&= \llbracket E^0 \rrbracket (\eta_{\llbracket \mathbf{real} \rrbracket}^\dagger(i[\mu]))(U) && \text{by Lemma B.2, item 2} \\
&= \llbracket E^0 \rrbracket (i[\mu])(U) && \text{since } \eta_X^\dagger = \text{id}_V X \\
&= \llbracket E^0[\mathbf{sample}[\mu]] \rrbracket (U) = \llbracket M \rrbracket (U).
\end{aligned}$$

If $M = E^0[\mathbf{score} \underline{a}]$, then:

$$\begin{aligned}
\text{Next}^{\leq n}(M)(\overline{U}) &= |\mathbf{a}|. \text{Next}^{\leq n-1}(E^0[\mathbf{ret} *])(\overline{U}) d\mu && \text{by Lemma 10.14, item 3} \\
&\leq |\mathbf{a}|. \llbracket E^0[\mathbf{ret} *] \rrbracket (U) && \text{by induction hypothesis} \\
&= |\mathbf{a}|. \llbracket E^0 \rrbracket (\delta_*)(U) \\
&= \llbracket E^0 \rrbracket (|\mathbf{a}|. \delta_*)(U) && \text{by Lemma B.2, item 1} \\
&= \llbracket E^0[\mathbf{score} \underline{a}] \rrbracket (U) = \llbracket M \rrbracket (U).
\end{aligned}$$

If $M = E[\underline{f} \ a_1 \ \cdots \ a_k]$, then we consider two cases. If $f(a_1) \cdots (a_k)$ is not observable, then $Next^{\leq n}(M)(\overline{U}) = 0$ by Lemma 10.14, item 4, and the claim is clear. If $f(a_1) \cdots (a_k)$ is observable, then:

$$\begin{aligned} Next^{\leq n}(M)(\overline{U}) &= Next^{\leq n-1}(E[f(a_1) \cdots (a_k)])(\overline{U}) && \text{by Lemma 10.14, item 4} \\ &\leq \llbracket E[f(a_1) \cdots (a_k)] \rrbracket (U) && \text{by induction hypothesis} \\ &= \llbracket E[\underline{f} \ a_1 \ \cdots \ a_k] \rrbracket (U) = \llbracket M \rrbracket (U). \end{aligned}$$

Finally, if M an instance L of a rule $L \rightarrow R$ of Figure 3 except (5), (6) and (7), then by inspection $\llbracket L \rrbracket = \llbracket R \rrbracket$, so:

$$\begin{aligned} Next^{\leq n}(M)(\overline{U}) &= Next^{\leq n-1}(R)(\overline{U}) && \text{by Lemma 10.14, item 5} \\ &\leq \llbracket R \rrbracket (U) && \text{by induction hypothesis} \\ &= \llbracket L \rrbracket (U) = \llbracket M \rrbracket (U). \end{aligned}$$

□

B.3 Adequacy (Section 10.2)

LEMMA B.3. *Let Σ consist of strictly observable first-order constants. For every string of rewrite steps $M = M_0 \rightarrow M_1 \rightarrow \cdots \rightarrow M_n \rightarrow \cdots$ by deterministic rules, $Next^*(M) = Next^*(M_i)$ for every i .*

PROOF. It suffices to show that if $M \rightarrow N$ by some deterministic rule, then $Next^*(M) = Next^*(N)$. This follows from the fact that, for every $n \geq 1$, $Next^{\leq n}(M) = Next^{\leq n-1}(N)$ (Lemma 10.14, items 4 and 5), and by taking suprema over n . □

LEMMA B.4. *Let Σ consist of strictly observable first-order constants.*

- (1) *For every generalized ISPCF term M of type τ , there is at most one maximal string of instances of weak deterministic rules $M_0 \stackrel{\text{def}}{=} M \rightarrow M_1 \rightarrow \cdots \rightarrow \cdots M_n \cdots$. Every M_i has type τ . If τ is an observable type, and if that string stops at rank n , then M_n is a weak value V^0 .*
- (2) *For every generalized ISPCF term M of type τ , there is at most one maximal string of instances of deterministic rules $M_0 \stackrel{\text{def}}{=} M \rightarrow M_1 \rightarrow \cdots \rightarrow \cdots M_n \cdots$. Every M_i has type τ . If τ is an observable type, either that string stops at rank n , then M_n is a value V , and $Next^*(M) = \delta_V$; or else $Next^*(M) = 0$.*

PROOF. The unicity of any maximal string of (weak) deterministic rules starting from M is clear, as well as the fact that every M_i has type τ . We now assume that τ is an observable type.

1. We consider a maximal string of weak deterministic rules stopping at rank n , namely $M = M_0 \rightarrow^* M_n$. We claim that M_n is weakly normal. If M_n is of the form $E^0[\mathbf{sample}[\mu]]$ or $E^0[\mathbf{score} \ \underline{a}]$ or $E^0[\mathbf{do} \ x \leftarrow \mathbf{ret} \ M; N]$, then E^0 has a type of the form $D\sigma \vdash \tau$. By Lemma B.1, that would imply that τ is of the form $D\tau'$ for some type τ' , which is impossible since τ is observable. Since no weak deterministic rule applies to M_n , no weak rule at all applies, so M_n is weakly normal. By Lemma 10.12, it is a weak value V^0 .

2. We reason similarly, assuming a maximal string of deterministic rules stopping at rank n , $M = M_0 \rightarrow^* M_n$. Then M_n is a value V . By Lemma B.3, $Next^*(M) = Next^*(M_n) = Next^*(V)$, and this is equal to δ_V by Lemma 10.14, item 1.

If instead the maximal string $M = M_0 \rightarrow M_1 \rightarrow \cdots$ does not stop, then for every $m \in \mathbb{N}$, we have $Next^{\leq m}(M) = Next^{\leq m-1}(M_1) = \cdots = Next^{\leq 0}(M_m)$, by Lemma 10.14, items 4 and 5. Since M_m is not normal, this is equal to 0. Taking suprema over $m \in \mathbb{N}$, we obtain that $Next^*(M) = 0$. □

LEMMA B.5. *Let Σ consist of strictly observable first-order constants. Let τ be any type. If $M \rightarrow^* M'$ by weak deterministic rules, then $M R_\tau a$ if and only if $M' R_\tau a$.*

PROOF. It suffices to show this when $M \rightarrow M'$ by one weak deterministic rule, and we show this by induction on τ .

When τ is a basic type, and if $M' R_\tau a$, then either $a = \perp$, or $M' \rightarrow^* \underline{b}$ by weak deterministic rules, with $b \geq a$. In the second case, $M \rightarrow M' \rightarrow^* \underline{b}$ by weak deterministic rules, so $M R_\tau a$. In the reverse direction, if $M R_\tau a$, either $a = \perp$, or $M \rightarrow^* \underline{b}$ by weak deterministic rules, with $b \geq a$. Since $M \rightarrow M'$ by a weak deterministic rule, since \underline{b} is (weakly) normal, and since weak deterministic rules are deterministic by construction, the string of reduction steps $M \rightarrow^* \underline{b}$ is of the form $M \rightarrow M' \rightarrow^* \underline{b}$. Therefore $M R_\tau a$.

The argument is similar when τ is a product type $\tau_1 \times \tau_2$. The key is that, if $M \rightarrow^* \langle M_1, M_2 \rangle$ by weak deterministic rules, and if $M \rightarrow M'$ by a weak deterministic rule, then M is not weakly normal, so the first string of rewrite steps must be of the form $M \rightarrow M' \rightarrow^* \langle M_1, M_2 \rangle$. Hence, if $M \rightarrow M'$ by a weak deterministic rule, then $M \rightarrow^* \langle M_1, M_2 \rangle$ by weak deterministic rules if and only if $M' \rightarrow^* \langle M_1, M_2 \rangle$ by weak deterministic rules. We reason similarly if τ is a sum type $\tau_1 + \tau_2$, where if $M \rightarrow M'$ by a weak deterministic rule, then $M \rightarrow^* \iota_i N$ by weak deterministic rules if and only if $M' \rightarrow^* \iota_i N$ by weak deterministic rules.

For arrow types $\sigma \rightarrow \tau$, we assume M and M' of type $\sigma \rightarrow \tau$, such that $M \rightarrow M'$ by a weak deterministic rule. Then, for every $N : \sigma$, $MN \rightarrow M'N$ by the same weak deterministic rule. If $M R_{\sigma \rightarrow \tau} f$, then for all $N R_\tau a$, $MN R_\tau f(a)$, and since $MN \rightarrow M'N$, by induction hypothesis $M'N R_\tau f(a)$; therefore $M' R_{\sigma \rightarrow \tau} f$. The fact that $M' R_{\sigma \rightarrow \tau} f$ implies $M R_{\sigma \rightarrow \tau} f$ is similar.

For distribution types $\tau \stackrel{\text{def}}{=} D\sigma$, we use a variant of that argument. We assume $M, M' : D\sigma$ such that $M \rightarrow M'$ by some weak deterministic rule. Then, for every weak evaluation context E^0 , $E^0[M] \rightarrow^* E^0[M']$ by the same weak deterministic rule. By Lemma B.3, $\text{Next}^*(E^0[M]) = \text{Next}^*(E^0[M'])$, from which it follows immediately that $M R_{D\sigma} \mu$ if and only if $M' R_{D\sigma} \mu$, for any μ . \square

LEMMA B.6. *Let Σ consist of strictly observable first-order constants. For every type τ , for every generalized ISPCF term M of type τ , $M R_\tau$ is a Scott-closed subset of $\llbracket \tau \rrbracket$ that contains \perp .*

PROOF. By induction on τ . That $M R_\tau$ contains \perp is clear except perhaps when τ is a distribution type $D\tau'$. In that case, the \perp element of $\llbracket \tau \rrbracket$ is the zero valuation 0 , and $h^\dagger(0)(U)$ is equal to 0 for every $h \in \llbracket \tau \vdash D\beta \rrbracket$, which trivially entails the statement.

If τ is a basic type, then $M R_\tau$ is the closed set $\{\perp\}$ if the maximal string of weak deterministic rewriting steps starting from M does not stop, or $\downarrow b$ if $M \rightarrow^* \underline{b}$ by weak deterministic rules. Then, $\downarrow b$ is closed because downward closures of points are closed.

In the case of product types, either $M \rightarrow^* V^0$ by weak deterministic rules for no lazy value V^0 , and then $M R_{\tau_1 \times \tau_2}$ is equal to $\{\perp\}$; or $M \rightarrow^* V^0$ for some uniquely determined lazy value V^0 by Lemma B.4, item 1. By typing and Lemma 10.12, $V^0 = \langle M_1, M_2 \rangle$ for some $M_1 : \tau_1$ and $M_2 : \tau_2$. Then $M R_{\tau_1 \times \tau_2}$ is equal to $(M_1 R_{\tau_1}) \times (M_2 R_{\tau_2})$ union $\{\perp\}$. The latter is a finite union of Scott-closed sets, hence is Scott-closed.

In the case of sum types, we reason similarly. Either $M R_{\tau_1 + \tau_2}$ is equal to $\{\perp\}$, or $M \rightarrow^* \iota_i N$ by weak deterministic rules, and then $M R_{\tau_1 + \tau_2}$ is the set of elements $\{\perp\} \cup \{(i, b) \mid b \in N R_{\tau_i}\}$. It is easy to see that the latter is downwards closed and closed under directed suprema, hence is Scott-closed.

In the case of arrow types $\sigma \rightarrow \tau$, $M R_{\sigma \rightarrow \tau}$ is equal to the intersection over all $N R_\sigma a$ of the sets $\text{App}(_, a)^{-1}(MN R_\tau)$. (App is the application morphism $(g, a) \mapsto g(a)$.) This set is closed because App and therefore $\text{App}(_, a)$ is continuous for every a , and intersections of closed sets are closed.

In the case of distribution types $D\tau$, $M R_{D\tau}$ is the intersection of the sets $(h^\dagger(_)(U))^{-1}(\downarrow \text{Next}^*(E^0[M])(\overline{U}))$ over all $E^0 R_{\tau \rightarrow D\beta}^\perp h$ and all observable open subsets U of $\llbracket \beta \rrbracket$. We conclude because downward closures of points are closed, and since h^\dagger is Scott-continuous, hence also $h^\dagger(_)(U)$ for every U . \square

LEMMA B.7. *Let Σ consist of strictly observable first-order constants. For every observable type β , for every observable element a of $\llbracket \beta \rrbracket$, $\underline{a} R_\beta a$.*

PROOF. We recall that \underline{a} was defined in Definition 10.4. We show this by induction on β . If β is a basic type, this follows from the fact that $\underline{a} \rightarrow^* a$ (in 0 step) and $\llbracket \underline{a} \rrbracket = a$. If β is a product type $\beta_1 \times \beta_2$, then a is of the form (a_1, a_2) with a_1 observable in $\llbracket \beta_1 \rrbracket$ and a_2 observable in $\llbracket \beta_2 \rrbracket$. Then $\underline{a} \rightarrow^* \langle \underline{a}_1, \underline{a}_2 \rangle$ (in 0 step), while $\underline{a}_1 R_{\beta_1} a_1$ and $\underline{a}_2 R_{\beta_2} a_2$ by induction hypothesis. The case of sum types is similar. \square

Proposition 10.20 (recap). *Let Σ consist of strictly observable first-order constants. For every ISPCF term M : τ , for all $\theta R_* \rho$, $M\theta R_\tau \llbracket M \rrbracket \rho$.*

PROOF. By induction on the size of M .

- If M is a constant \underline{f} , say of type $\sigma_1 \rightarrow \dots \rightarrow \sigma_k \rightarrow \tau$ where every σ_i is a basic type and τ is an observable type, then we must show that for all $N_1 R_{\sigma_1} a_1, \dots, N_k R_{\sigma_k} a_k$, $\underline{f} N_1 \dots N_k R_\tau f(a_1) \dots (a_k)$. If $f(a_1) \dots (a_k) = \perp$, this is clear, so let us assume $f(a_1) \dots (a_k) \neq \perp$. By the second condition of Definition 10.5, $f(a_1) \dots (a_k)$ is observable, and every a_i is different from \perp . Since every σ_i is a basic type, we have $N_i \rightarrow^* \underline{a}'_i$ by weak deterministic rules for some zero-ary constant \underline{a}'_i such that $\underline{a}'_i \geq a_i$, for every i with $1 \leq i \leq k$. It is easy to see that $\underline{f} N_1 \dots N_k \rightarrow^* \underline{f} \underline{a}'_1 \dots \underline{a}'_k \rightarrow \underline{f}(\underline{a}'_1) \dots (\underline{a}'_k)$ by weak deterministic rules; the last step is justified by the fact that $f(\underline{a}'_1) \dots (\underline{a}'_k) \geq f(a_1) \dots (a_k)$, that $\underline{f}(\underline{a}'_1) \dots (\underline{a}'_k)$ is observable, and that every element larger than an observable element is itself observable. By Lemma B.7, $\underline{f}(\underline{a}'_1) \dots (\underline{a}'_k) R_\tau \llbracket \underline{f}(\underline{a}'_1) \dots (\underline{a}'_k) \rrbracket \rho = f(\underline{a}'_1) \dots (\underline{a}'_k)$. Using Lemma B.5, we deduce that $\underline{f} N_1 \dots N_k R_\tau f(\underline{a}'_1) \dots (\underline{a}'_k)$. Since $f(\underline{a}'_1) \dots (\underline{a}'_k) \geq f(a_1) \dots (a_k)$, and since $\underline{f} N_1 \dots N_k R_\tau$ is Scott-closed, hence downwards-closed (Lemma B.6), we obtain that $\underline{f} N_1 \dots N_k R_\tau f(a_1) \dots (a_k)$, as desired.

- If M is a variable, this is by the assumption $\theta R_* \rho$.

- If $M = \mathbf{sample}[\mu]$, we must show that for all $E^0 R_{\mathbf{real} \rightarrow D\beta}^\perp h$, for every observable open subset U of $\llbracket \beta \rrbracket$, $\text{Next}^*(E^0[\mathbf{sample}[\mu]])(\overline{U}) \geq h^\dagger(\mu)(U)$. By Lemma 10.14, item 2, and taking suprema over $n \geq 1$, $\text{Next}^*(E^0[\mathbf{sample}[\mu]])(\overline{U})$ is equal to $\int_{a \in \mathbb{R}} \text{Next}^*(E^0[\mathbf{ret} \underline{a}])(\overline{U}) d\mu$. By Lemma B.7, $\underline{a} R_{\mathbf{real}} a$ for every $a \in \mathbb{R}$. In particular, $\underline{a} R_{\mathbf{real}} i(a)$ for every $a \in \mathbb{R}$. Since $E^0 R_{\mathbf{real} \rightarrow D\beta}^\perp h$, $\text{Next}^*(E^0[\mathbf{ret} \underline{a}])(\overline{U}) \geq h(a)(U)$ for every $a \in \mathbb{R}$. From this, we conclude that $\text{Next}^*(E^0[\mathbf{sample}[\mu]])(\overline{U}) \geq \int_{a \in \mathbb{R}} h(a)(U) d\mu = h^\dagger(\mu)(U)$.

- If $M = \mathbf{score} N$, we must show that for all $E^0 R_{\mathbf{unit} \rightarrow D\beta}^\perp h$, for every observable open subset U of $\llbracket \beta \rrbracket$, $\text{Next}^*(E^0[M\theta])(\overline{U}) \geq h^\dagger(|a|. \delta_*)(U)$ where $\mathbf{a} \stackrel{\text{def}}{=} \llbracket N \rrbracket \rho$. We observe that $h^\dagger(|a|. \delta_*) = |a|.h(*)$. Hence the claim is clear if $|a| = 0$, notably if $\mathbf{a} = \perp$. Let us assume that $\mathbf{a} \neq \perp$. Then $N\theta R_{\mathbf{real}} \mathbf{a}$, by induction hypothesis. By definition of $R_{\mathbf{real}}$, $N\theta \rightarrow^* \underline{a}'$ by weak deterministic rules, for some zero-ary constant \underline{a}' such that $\mathbf{a}' \geq \mathbf{a}$.

Since $E^0 R_{\mathbf{unit} \rightarrow D\beta}^\perp h$, and since $* R_{\mathbf{unit}} *$ (by the case of zero-ary constants, already treated), $\text{Next}^*(E^0[\mathbf{ret} *])(\overline{U}) \geq h(*) (U)$. We have $N\theta \rightarrow^* \underline{a}'$ by weak deterministic rules, so $E^0[M\theta] = E^0[\mathbf{score} N\theta] \rightarrow^* E^0[\mathbf{score} \underline{a}']$ by weak deterministic rules. By Lemma B.3, $\text{Next}^*(E^0[M\theta])(\overline{U}) = \text{Next}^*(E^0[\mathbf{score} \underline{a}'])(\overline{U})$, which is equal to $|a'|. \text{Next}^*(E^0[\mathbf{ret} *])(\overline{U})$ by Lemma 10.14, item 3, hence is larger than or equal to $|a'|.h(*) (U)$. We verify that $|a'| \geq |a|$, owing to the fact that $\mathbf{a}' \geq \mathbf{a}$. Hence $\text{Next}^*(E^0[M\theta])(\overline{U}) \geq |a|.h(*) (U) = h^\dagger(|a|. \delta_*) (U)$.

• If $M = \lambda x.N: \sigma \rightarrow \tau$, then we must show that for all $P R_\sigma a$, $(M\theta)P R_\tau \llbracket M \rrbracket \rho(a)$. By induction hypothesis, we have $N\theta[x := P] R_\tau \llbracket N \rrbracket \rho[x \mapsto a]$. Since $(M\theta)P \rightarrow N\theta[x := P]$, we conclude by Lemma B.5.

• The case where M is an application is immediate.

• If $M = \mathbf{rec} N$, where $N: \sigma \rightarrow \sigma$, then let $f \stackrel{\text{def}}{=} \llbracket N \rrbracket \rho$. By induction hypothesis, $N R_{\sigma \rightarrow \sigma} f$, so: (*) for all $P R_\sigma a$, we have $(N\theta)P R_\sigma f(a)$. We show that $M\theta R_\sigma f^n(\perp)$ for every $n \in \mathbb{N}$, by induction on n . If $n = 0$, then $f^0(\perp) = \perp$ is in $M\theta R_\sigma$ by Lemma B.6. If $n \geq 1$, then $N\theta(M\theta) R_\sigma f(f^{n-1}(\perp))$, using (*) and the induction hypothesis. Now $M\theta = \mathbf{rec}(N\theta) \rightarrow N\theta(M\theta)$ by a weak deterministic rule, so $M\theta R_\sigma f^n(\perp)$, by Lemma B.5. Now that we have shown that $M\theta R_\sigma f^n(\perp)$, namely that $f^n(\perp)$ is in $M\theta R_\sigma$ for every $n \in \mathbb{N}$, we use the fact that $M\theta R_\sigma$ is Scott-closed (Lemma B.6) and we conclude that $\llbracket M \rrbracket \rho = \text{lfp}(f) = \sup_{n \in \mathbb{N}} f^n(\perp)$ is in $M\theta R_\sigma$.

• If $M = \mathbf{ret} N$, with $N: \sigma$, then we must show that for all $E^0 R_{\sigma+D\beta}^\perp h$, for every observable open subset U of $\llbracket \beta \rrbracket$, $\text{Next}^*(E^0[M\theta])(\bar{U}) \geq h^\dagger(\delta_{\llbracket N \rrbracket \rho})(U)$. By induction hypothesis, $N\theta R_\sigma \llbracket N \rrbracket \rho$, so by definition of $R_{\sigma+D\beta}^\perp$, we obtain $\text{Next}^*(E^0[\mathbf{ret} N\theta])(\bar{U}) \geq h(\llbracket N \rrbracket \rho)(U)$; this is exactly what we want to prove.

• If $M = \mathbf{do} x_\sigma \rightarrow N; P: D\tau$, then we must show that for all $E^0 R_{\tau+D\beta}^\perp h$, for every observable open subset U of $\llbracket \beta \rrbracket$, $\text{Next}^*(E^0[M\theta])(\bar{U}) \geq h^\dagger(\llbracket M \rrbracket \rho)(U)$. We note that $h^\dagger(\llbracket M \rrbracket \rho)(U) = h^\dagger((\llbracket \lambda x_\sigma.P \rrbracket \rho)^\dagger(\llbracket N \rrbracket \rho))(U) = (h^\dagger \circ \llbracket \lambda x_\sigma.P \rrbracket \rho)^\dagger(\llbracket N \rrbracket \rho)(U)$, where the last equality is by the monad equation $g^\dagger \circ f^\dagger = (g^\dagger \circ f)^\dagger$.

We consider the weak evaluation context $E'^0 \stackrel{\text{def}}{=} E^0[\mathbf{do} x_\sigma \rightarrow \cdot; P\theta]$, and we claim that $E'^0 R_{\sigma+D\beta}^\perp h^\dagger \circ \llbracket \lambda x_\sigma.P \rrbracket \rho$. This means showing that for all $Q R_\sigma b$, for every observable open subset V of $\llbracket D\beta \rrbracket$, $\text{Next}^*(E'^0[\mathbf{ret} Q])(\bar{V}) \geq h^\dagger(\llbracket \lambda x_\sigma.P \rrbracket \rho(b))(V)$. The right-hand side is equal to $h^\dagger(\llbracket P \rrbracket \rho[x \mapsto b])(V)$. Since $E'^0[\mathbf{ret} Q] = E^0[\mathbf{do} x_\sigma \rightarrow \mathbf{ret} Q; P\theta] \rightarrow E^0[P\theta[x := Q]]$ by a weak deterministic rule, and by Lemma B.3, the left-hand side $\text{Next}^*(E'^0[\mathbf{ret} Q])(\bar{V})$ is equal to $\text{Next}^*(E^0[P\theta[x := Q]])(\bar{V})$. The latter is larger than or equal to $h^\dagger(\llbracket P \rrbracket \rho[x \mapsto b])(V)$ since $P\theta[x := Q] R_{D\tau} \llbracket P \rrbracket \rho[x \mapsto b]$ by induction hypothesis and since $E^0 R_{\tau+D\beta}^\perp h$.

Since $E'^0 R_{\sigma+D\beta}^\perp h^\dagger \circ \llbracket \lambda x_\sigma.P \rrbracket \rho$ and since by induction hypothesis $N\theta R_{D\sigma} \llbracket N \rrbracket \rho$, we have $\text{Next}^*(E'^0[N\theta])(\bar{U}) \geq (h^\dagger \circ \llbracket \lambda x_\sigma.P \rrbracket \rho)^\dagger(\llbracket N \rrbracket \rho)(U)$, and this is exactly what we had to prove.

• If $M = \langle M_1, M_2 \rangle$, with $M_1: \tau_1$ and $M_2: \tau_2$, then $M\theta \rightarrow^* \langle M_1\theta, M_2\theta \rangle$ by weak deterministic rules (vacuously), and $M_1\theta R_{\tau_1} \llbracket M_1 \rrbracket \rho$, $M_2\theta R_{\tau_2} \llbracket M_2 \rrbracket \rho$, by induction hypothesis.

• If $M = \pi_1 N$, with $N: \tau_1 \times \tau_2$, then $N\theta R_{\tau_1 \times \tau_2} \llbracket N \rrbracket \rho$ by induction hypothesis. If $\llbracket N \rrbracket \rho = \perp$, then $\llbracket M \rrbracket \rho = \perp$, and therefore $M\theta R_{\tau_1 \times \tau_2} = \llbracket M \rrbracket \rho$, by Lemma B.6. Otherwise, $\llbracket N \rrbracket \rho$ is of the form (a_1, a_2) with $a_1 \neq \perp$ or $a_2 \neq \perp$. By definition of $R_{\tau_1 \times \tau_2}$, $N\theta \rightarrow^* \langle N_1, N_2 \rangle$ by weak deterministic rules in such a way that $N_1 R_{\tau_1} a_1$ and $N_2 R_{\tau_2} a_2$. We note that $M\theta = \pi_1 N\theta \rightarrow^* \pi_1 \langle N_1, N_2 \rangle \rightarrow N_1$ by weak deterministic rules, so $M\theta R_{\tau_1} a_1 = \llbracket M \rrbracket \rho$, using Lemma B.5.

The case $M = \pi_2 N$ is similar.

• If $M = \iota_1 N: \tau_1 + \tau_2$, then vacuously $M\theta \rightarrow^* \iota_1 N\theta$ by weak deterministic rules. By induction hypothesis, $N\theta R_{\tau_1} \llbracket N \rrbracket \rho$, so $M\theta R_{\tau_1 + \tau_2} (\iota_1, \llbracket N \rrbracket \rho) = \llbracket M \rrbracket \rho$.

The case $M = \iota_2 N$ is similar.

• If $M = \mathbf{case} NP_1 P_2: \tau$, with $N: \tau_1 + \tau_2$, then by induction hypothesis $N\theta R_{\tau_1 + \tau_2} \llbracket N \rrbracket \rho$. If $\llbracket N \rrbracket \rho = \perp$, then $\llbracket M \rrbracket \rho = \perp$, so $M\theta R_\tau \llbracket M \rrbracket \rho$, by Lemma B.6. Otherwise, $\llbracket N \rrbracket \rho = (i, b)$ for some $i \in \{1, 2\}$ and $b \in \llbracket \tau_i \rrbracket$. By definition of $R_{\tau_1 + \tau_2}$, $N\theta \rightarrow^* \iota_i N'$ by weak deterministic rules, in such a way that $N' R_{\tau_i} b$. Then $M\theta \rightarrow^* \mathbf{case}(\iota_i N')(P_1\theta)(P_2\theta) \rightarrow P_i\theta N'$ by weak deterministic rules. By induction hypothesis, $P_i\theta R_{\tau_i \rightarrow \tau} \llbracket P_i \rrbracket \rho$, so $P_i\theta N' R_\tau \llbracket P_i \rrbracket \rho(b)$. By Lemma B.5, $M\theta R_\tau \llbracket P_i \rrbracket \rho(b) = \llbracket M \rrbracket \rho$. \square

Lemma 10.22 (recap). *Let Σ consist of strictly observable first-order constants. For every observable type β , for every generalized ISPCF term $N : \beta$, for every $b \in \llbracket \beta \rrbracket$, if $N R_\beta b$ then either b is not observable, or b is observable and $N \rightarrow^* V$ by deterministic rules for some value $V : \beta$ such that $\llbracket V \rrbracket \geq b$.*

PROOF. We show that $N R_\beta b$ and the fact that b is observable together imply the existence of a value $V : \beta$ such that $\llbracket V \rrbracket \geq b$, by induction on R_β . If β is a basic type, then this is by definition of R_β . If β is a product type $\beta_1 \times \beta_2$, then let us write b as (b_1, b_2) , where $b_1 \neq \perp$ and $b_2 \neq \perp$ since b is observable. Since $b \neq \perp$, by definition of $R_{\beta_1 \times \beta_2}$, $M \rightarrow^* \langle M_1, M_2 \rangle$ by weak deterministic rules, for some $M_1 R_{\tau_1} b_1$ and $M_2 R_{\tau_2} b_2$. By induction hypothesis, $M_1 \rightarrow^* V_1$ and $M_2 \rightarrow^* V_2$ by weak deterministic rules, for some values V_1 and V_2 such that $\llbracket V_1 \rrbracket \geq b_1$, $\llbracket V_2 \rrbracket \geq b_2$. Then $M \rightarrow^* \langle M_1, M_2 \rangle \rightarrow^* \langle V_1, M_2 \rangle \rightarrow^* \langle V_1, V_2 \rangle$ by (non-weak) deterministic rules. The term $V \stackrel{\text{def}}{=} \langle V_1, V_2 \rangle$ is a value, and $\llbracket V \rrbracket \geq (b_1, b_2) = b$. The case of sum types is similar. \square

We can now establish adequacy. **Theorem 10.23 (recap).** *Let Σ consist of strictly observable first-order constants. For every observable type β , for every generalized ISPCF term $M : D\beta$, for every observable open subset U of $\llbracket \beta \rrbracket$, $\llbracket M \rrbracket (U) = \text{Next}^*(M)(\overline{U})$.*

PROOF. We consider the empty weak evaluation context $E^0 \stackrel{\text{def}}{=} []$, and we claim that $E^0 R_{\beta \vdash D\beta}^\perp \eta_{\llbracket \beta \rrbracket}$. In order to show this, we need to show that for all $N R_\beta b$, for every observable open subset U of $\llbracket \beta \rrbracket$, $\text{Next}^*(E^0[\mathbf{ret} N])(\overline{U}) \geq \delta_b(U)$.

If $b \notin U$, then $\delta_b(U) = 0$, and this is obvious.

Otherwise, $\delta_b(U) = 1$. Since $b \in U$ and since U is observable, b is an observable element of $\llbracket \beta \rrbracket$. Since $N R_\beta b$, Lemma 10.22 tells us that $N \rightarrow^* V$ by deterministic rules for some value $V : \beta$ such that $\llbracket V \rrbracket \geq b$. Since U is upwards closed, $\llbracket V \rrbracket$ is in U . Therefore V is in \overline{U} , and $\mathbf{ret} V$ is in \overline{U} .

We note that since $N \rightarrow^* V$ by deterministic rules, $\mathbf{ret} N \rightarrow^* \mathbf{ret} V$ by deterministic rules as well. By Lemma B.3, $\text{Next}^*(E^0[\mathbf{ret} N]) = \text{Next}^*(\mathbf{ret} N)$ is equal to $\text{Next}^*(\mathbf{ret} V)$. By Lemma 10.14, item 1, (and taking suprema over all n), since $\mathbf{ret} V$ is a normal form, $\text{Next}^*(\mathbf{ret} V)(\overline{U}) = \delta_{\mathbf{ret} V}(\overline{U}) = 1 = \delta_b(U)$.

Now that we have proved $E R_{\beta \vdash D\beta}^\perp \eta_{\llbracket \beta \rrbracket}$, we apply it to $M R_{D\beta} \llbracket M \rrbracket$ (Corollary 10.21), and, noting that $E = []$ is a weak evaluation context, we obtain that $\text{Next}^*(E[M])(\overline{U}) \geq (\eta_{\llbracket \beta \rrbracket})^\dagger(\llbracket M \rrbracket)(U) = \llbracket M \rrbracket (U)$. The converse inequality is by soundness (Proposition 10.17). \square

B.4 The sampling-based operational semantics (Section 12.2)

Lemma 12.5 (recap). *Under the assumptions of Definition 12.1, for every open subset U of Norm, the maps $\sigma_k[U]$ ($k \in \mathbb{N}$) and $\sigma[U]$ are lower semicontinuous from $\Gamma \times \{0, 1\}^{\mathbb{N}}$ to \mathbb{R}_+ .*

PROOF. We recall that lower semicontinuous simply means continuous, provided that we equip \mathbb{R}_+ with its Scott topology, which we now assume.

In order to simplify the proof, we use the following result by Ershov [18, Proposition 2]: given a c-space X (an α -space in Ershov's terminology, e.g., a continuous poset with its Scott topology), and two topological spaces Y and Z , a function $f : X \times Y \rightarrow Z$ is separately continuous if and only if it is jointly continuous. Joint continuity is ordinary continuity from $X \times Y$ with the product topology. Separate continuity means that $f(x, _)$ and $f(_, y)$ are continuous for fixed x and y respectively. This applies here, as Γ' is a continuous poset.

Since suprema of lower semicontinuous maps are lower semicontinuous, it suffices to show that $\sigma_k[U]$ is lower semicontinuous for every $k \in \mathbb{N}$, which we do by induction on k .

We fix a template M_0 , and we show that $\sigma_k[U]$ is continuous from the product of the space of generalized ISPCF terms of shape M_0 with $\{0, 1\}^{\mathbb{N}}$ to $\overline{\mathbb{R}}_+$. We only need to prove separate continuity.

If $k = 0$, then $\sigma_0[U](M, s) = 1$ if M is normal, 0 otherwise. For M fixed, this is constant hence continuous in s . For s fixed, we note that normality only depends on the shape (here, M_0). Hence $\sigma_0[U](_, s)$ is constant, hence continuous, on the subspace of generalized ISPCF terms of shape M_0 .

Let us assume $k \geq 1$. If the unique \rightarrow_s step from (M, i, r) is by applying a rule $L \rightarrow R$ of $L \rightarrow R$ of Figure 3 except (5), (6) and (7), then we note that the same rule would apply if we replaced M by any other term with the same shape M_0 . Then $\sigma_k[U](M, s) = \sigma_{k-1}[U](R, s)$, and we apply the induction hypothesis.

In the case of **sample**, $M = E[\mathbf{sample}[\lambda_1]]$, and if $(M, 0, 1) \xrightarrow{\leq k} (M', i', r')$ for some $M' \in U$, then $(M, 0, 1) \rightarrow_s (E[\mathbf{ret\ num}(s[0])], 1, 1) \xrightarrow{\leq k-1} (M', i', r')$, where $r' = \sigma_k[U](M, s)$. We then have $(E[\mathbf{ret\ num}(s[0])], 0, 1) \xrightarrow{\leq k-1} (M', i' - 1, r')$ by Lemma 12.3, hence $r' = \sigma_k[U](M, \uparrow s) = \sigma_{k-1}[U](E[\mathbf{ret\ num}(s[0])], \uparrow s)$. For M fixed, we see that the composition of the maps $s \mapsto s[0] \mapsto \mathbf{num}(s[0])$ is continuous from $\{0, 1\}^{\mathbb{N}}$ to \mathbb{R} , using Lemma 9.3; also, $s \mapsto \uparrow s$ is continuous. As in the proof of Proposition 10.8, the map $a \in \mathbb{R} \mapsto E[\mathbf{ret\ a}]$ is continuous, so $\sigma_k[U](M, s)$ defines a continuous function of s . For s fixed, this is the composition of the continuous function that maps every term $M = E[\mathbf{sample}[\lambda_1]]$ of shape M_0 to $E[\mathbf{ret\ a}]$ (where $a \stackrel{\text{def}}{=} \mathbf{num}(s[0])$) with $\sigma_{k-1}[U](_, \uparrow s)$, hence is continuous as well.

In the case of **score**, we have $M = E[\mathbf{score\ a}]$, and assuming that $(M, 0, 1) \xrightarrow{\leq k} (M', i', r')$ for some $M' \in U$, we have $(M, 0, 1) \rightarrow_s (E[\mathbf{ret\ *}], 0, |a|) \xrightarrow{\leq k-1} (M', i', r')$, and $r' = \sigma_k[U](M, s)$. In particular, if $|a| \neq 0$ then $(E[\mathbf{ret\ *}], 0, 1) \xrightarrow{\leq k-1} (M', i', r'/|a|)$ by Lemma 12.4 (final part), so $\sigma_{k-1}[U](E[\mathbf{ret\ *}], s) = r'/|a|$. Therefore $\sigma_k[U](M, s) = |a| \cdot \sigma_{k-1}[U](E[\mathbf{ret\ *}], s)$. We see that this holds also when $|a| = 0$, using the first part of Lemma 12.4, and also when $(M, 0, 1) \xrightarrow{\leq k} (M', i', r')$ for no $M' \in U$, in which case both sides of the equality are 0. By induction hypothesis, σ_{k-1} is continuous. The map $M = E[\mathbf{score\ a}] \mapsto E[\mathbf{ret\ *}]$ is continuous, by a similar argument as in Proposition 10.8. The map $M = E[\mathbf{score\ a}] \mapsto a$ is also clearly continuous, the map $a \mapsto |a|$ is continuous from $\mathbb{I}\mathbb{R}_\perp$ to $\overline{\mathbb{R}}_+$, and product is Scott-continuous on $\overline{\mathbb{R}}_+$, so $\sigma_k[U](_, s)$ is continuous, for every fixed s . For $M = E[\mathbf{score\ a}]$ fixed, $\sigma_k[U](M, _) = |a| \cdot \sigma_{k-1}[U](E[\mathbf{ret\ *}], _)$ is obviously continuous. Hence $\sigma_k[U]$ itself is continuous.

In case $M = E[\underline{f\ a_1} \cdots \underline{a_k}]$ where $k = \alpha(\underline{f}) \neq 0$, then either $f(a_1) \cdots (a_k)$ is observable and $\sigma_k[U](M, s) = \sigma_{k-1}[U](E[\underline{f(a_1)} \cdots (a_k)], s)$, or else $\sigma_k[U](M, s) = 0$. For M fixed, this defines a continuous map in s . We now consider s fixed. The function that maps every term M as above, of shape M_0 , to the tuple of (denotations of) constants of type **real** among $\underline{a_1}, \dots, \underline{a_k}$ is continuous, hence also the function that maps M (of given shape M_0) to $f(a_1) \cdots (a_k)$. In particular the set V of generalized ISPCF terms M of shape M_0 such that $f(a_1) \cdots (a_k)$ is observable is open. Let g be the function that maps every $M = E[\underline{f\ a_1} \cdots \underline{a_k}]$ in V to $E[\underline{f(a_1)} \cdots (a_k)]$. This is continuous. The inverse image of every non-trivial Scott-open subset $]a, +\infty[$ of $\overline{\mathbb{R}}_+$ (namely, $a > 0$) by $\sigma_k[U](_, s)$ is equal to $g^{-1}(\sigma_{k-1}[U](_, s)^{-1}]a, +\infty[$ is then open, showing that $\sigma_k[U](_, s)$ is continuous. \square

Proposition 12.6 (recap). *Assume that all constants in Σ are first-order constants, and that the only available term of the form $\mathbf{sample}[\mu]$ is $\mathbf{sample}[\lambda_1]$, where λ_1 is Lebesgue measure on $[0, 1]$. For every open subset U of Norm, for every generalized ISPCF term M , $P[M \rightarrow U] = \mathit{Next}^*(M)(U)$.*

PROOF. We show that $\int_{s \in \{0, 1\}^{\mathbb{N}}} \sigma_k[U](M, s) dv = \mathit{Next}^{\leq k}(M)(U)$ by induction on $k \in \mathbb{N}$. The result will follow by taking suprema over k .

When $k = 0$, $\mathit{Next}^{\leq 0}(M)(U)$ is equal to $\delta_M(U)$ if $M \in \mathit{Norm}$, to 0 otherwise, hence to $\chi_U(M)$ in all cases. For every $s \in \{0, 1\}^{\mathbb{N}}$, $\sigma_0[U](M, s)$ is equal to $\chi_U(M)$, and since v is a probability distribution, $\int_{s \in \{0, 1\}^{\mathbb{N}}} \sigma_0[U](M, s) dv = \chi_U(M)$.

When $k \geq 1$, we look at the shape of M , and we use Lemma 10.14 to evaluate $\mathit{Next}^{\leq k}(M)(U)$.

If $M \in \text{Norm}$, then $\text{Next}^{\leq k}(M)(U) = \delta_M(U) = \chi_U(M)$. Also, $\sigma_k[U](M) = \chi_U(M)$, so $\int_{s \in \{0,1\}^{\mathbb{N}}} \sigma_k[U](M, s) \, dv = \chi_U(M)$.

We arrive at the main case of the proof. If $M = E[\mathbf{sample}[\mu]]$, where necessarily $\mu = \lambda_1$, then, as in the proof of Lemma 12.5, $\sigma_k[U](M, \uparrow s) = \sigma_{k-1}[U](E[\mathbf{ret \underline{num}(s[0])}], \uparrow s)$, so:

$$\int_{s \in \{0,1\}^{\mathbb{N}}} \sigma_k[U](M, s) \, dv = \int_{s \in \{0,1\}^{\mathbb{N}}} \sigma_{k-1}[U](E[\mathbf{ret \underline{num}(s[0])}], \uparrow s) \, dv.$$

We now use the change-of-variables formula with respect to the homeomorphism $\text{split}: s \in \{0,1\}^{\mathbb{N}} \mapsto (s[0], \uparrow s) \in (\{0,1\}^{\mathbb{N}})^2$ (see Lemma 9.4). Since $\text{split}[v] = v \otimes v$, and using Proposition 4.1:

$$\begin{aligned} \int_{s \in \{0,1\}^{\mathbb{N}}} \sigma_k[U](M, s) \, dv &= \int_{t \in \{0,1\}^{\mathbb{N}}} \left(\int_{s' \in \{0,1\}^{\mathbb{N}}} \sigma_{k-1}[U](E[\mathbf{ret \underline{num}(t)}], s') \, dv \right) \, dv \\ &= \int_{t \in \{0,1\}^{\mathbb{N}}} \text{Next}^{\leq k-1}(E[\mathbf{ret \underline{num}(t)}])(U) \, dv \\ &\quad \text{by induction hypothesis} \\ &= \int_{a \in \mathbb{R}} \text{Next}^{\leq k-1}(E[\mathbf{ret \underline{a}}])(U) \, d\lambda_1 \\ &\quad \text{by the change-of-variables formula and Lemma 9.3} \\ &= \text{Next}^{\leq k}(M)(U), \end{aligned}$$

by Lemma 10.14, item 2.

If $M = E[\mathbf{score \underline{a}}]$, then, as in the proof of Lemma 12.5, $\sigma_k[U](M, s) = |\mathbf{a}| \cdot \sigma_{k-1}[U](E[\mathbf{ret \underline{*}}], s)$. Hence:

$$\begin{aligned} \int_{s \in \{0,1\}^{\mathbb{N}}} \sigma_k[U](M, s) \, dv &= \int_{s \in \{0,1\}^{\mathbb{N}}} |\mathbf{a}| \cdot \sigma_{k-1}[U](E[\mathbf{ret \underline{*}}], s) \, dv \\ &= |\mathbf{a}| \cdot \text{Next}^{\leq k-1}(E[\mathbf{ret \underline{*}}], s)(U) = \text{Next}^{\leq k}(M)(U), \end{aligned}$$

by induction hypothesis and Lemma 10.14, item 3. The remaining cases are equally easy, and left to the reader. \square