Simply-Typed Measurability

OHAD KAMMAR, University of Edinburgh, Scotland, UK

We use simple types to structure complex measurability proofs. We interpret these types as the quasi-Borel spaces of Staton et al., and programs as quasi-measurable functions between them. In particular, we can express complex measurability proofs as Boolean programs that refine a simply-typed space into a measurable subset of interest. These spaces conservatively include the concrete spaces: the standard Borel spaces. As a consequence, reasoning about a concrete space will prove that the same subsets are measurable, but quasi-Borel spaces may give a simpler or shorter proof that involves intermediate non-concrete spaces. We apply this technique to recover the following well-known concrete spaces as subspaces of the full quasi-Borel function-space: the space of continuous functions and the Skorokhod space of right-continuous with left-limits (rcll/càdlàg) functions.

CCS Concepts: • **Mathematics of computing** → *Bayesian nonparametric models*; Point-set topology; **Lambda calculus**; • **Theory of computation** → **Denotational semantics**; *Program specifications*; **Higher order logic**; **Type theory**; Hoare logic; Constructive mathematics; *Logic and verification*; **Type structures**; **Functional constructs**.

ACM Reference Format:
Ohad Kammar. 2024. Simply-Typed Measurability. *ACM Trans. Probab. Mach. Learn.* 1, 1, Article 1 (January 2024), 45 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Probabilistic programming languages and their semantic theories marry the discrete and continuous worlds. On the one hand, they involve logical and typically discrete structures, such as programs, compilers, types, etc. On the other hand, they involve typically continuous concepts—in the sense of the real-number line and the continuum: diffuse probability distributions like the uniform distribution on the unit interval or Gaussians over the real line. Thus reasoning about probabilistic programs is tightly related to reasoning about their limiting behaviour which may involve continuous spaces. To account for continuous probability, we need to restrict attention to measurable events and measurable random variables. Measurable spaces and functions are the classical foundation for these concepts. Measurable spaces support many typical constructions such as: products, disjoint unions, inductively-defined relations, subspaces, etc. However, measurable spaces do not support all operations of interest, for example, we cannot have a measurable space of all random variables or all measurable events over the real line [Aumann 1961].

Probabilistic programming language semantics exacerbate the need for an extensive measurability toolkit. It may be straightforward to prove that a specific model of interest, i.e., a program, is measurable. However, a semantics for an entire probabilistic programming language require us to show that all expressible models are measurable. If the semantics requires sophisticated structure, we need to guarantee that these semantics will preserve measurability. A compelling idea is to

Author's address: Ohad Kammar, ohad.kammar@ed.ac.uk, University of Edinburgh, Laboratory for the Foundations of Computer Science, School of Informatics, 10 Crichton Street, Edinburgh, Scotland, UK, EH8 9AB.

ACM 2836-8924/2024/1-ART1
https://doi.org/XXXXXXX.XXXXXXX

define measurable spaces for the intermediate semantic structures. However, no-go results such as Aumann's theorem renders this idea dead in the water.

Recent years have seen various breakthroughs in the semantics of probabilistic programming that support such spaces [Bacci et al. 2018; Dahlqvist and Kozen 2020; Ehrhard et al. 2018; Goubault-Larrecq et al. 2023; Jia et al. 2021, e.g.].These kinds of semantic domains provide a meta-theory for probabilistic programs which supports mathematical constructions that can express programmable modelling operations of interest such as higher-order functions. Here we focus on a secondary use of such semantic domains: they enable a type-rich meta-language. Concretely, we investigate how one such model, the *quasi-Borel spaces* of Staton et al. [2017], to abstract away from the relatively low-level and technical details of measurability. The richer type-structure allows us to develop a modular collection of abstraction that reduce measurability proofs to their reusable building blocks that we can discharge by mere type-checking.

For example, the classical measure theoretic account supports a measurable function with the following type. It satisfies the specification on the right, i.e., it sends a real number $r \in \mathbb{R}$ to a sequence of rational numbers $((\mathsf{approx}\, r)_n)_{n\in\mathbb{N}}$ approximating it from below:

$$\mathsf{approx} : \mathbb{R} \to \mathbb{Q}^{\mathbb{N}} \qquad \forall r \in \mathbb{R}, n \in \mathbb{N}. r - \frac{1}{n} < (\mathsf{approx}\, r)_n < r$$

We can use it to guarantee a quadratic convergence rate:

$$\mathsf{approxFaster} : \mathbb{R} \to \mathbb{Q}^{\mathbb{N}} \qquad \mathsf{approxFaster}\, r \coloneqq ((\mathsf{approxFaster}\, r)_{n^2})_{n\in\mathbb{N}}$$

Here we can reduce the measurability argument to an abstract type-checking argument:

- Terms-in-contexts represent measurable functions, e.g.:

$$m : \mathbb{N}, n : \mathbb{N} \vdash m \cdot n : \mathbb{N} \quad \text{represents multiplication} \quad \mathbb{N} \times \mathbb{N} \xrightarrow{(\cdot)} \mathbb{N}$$

  and so, for example, we have the following well-formed terms:

$$r : \mathbb{R}, n : \mathbb{N} \vdash \mathsf{approx}\, r : \mathbb{Q}^{\mathbb{N}} \qquad r : \mathbb{R}, n : \mathbb{N} \vdash n^2 : \mathbb{N}$$

- The measurable space $\mathbb{Q}^{\mathbb{N}}$ supports two term-formers:

$$\frac{n : \mathbb{N} \vdash M : \mathbb{Q}}{\vdash (M)_{n\in\mathbb{N}} : \mathbb{Q}^{\mathbb{N}}} \qquad \frac{\vdash N : \mathbb{Q}^{\mathbb{N}} \qquad \vdash K : \mathbb{N}}{\vdash N_K : \mathbb{Q}}$$

  For example, we can form the following terms:

$$r : \mathbb{R}, n : \mathbb{N} \vdash (\mathsf{approx}\, r)_{n^2} : \mathbb{Q} \qquad r : \mathbb{R} \vdash ((\mathsf{approx}\, r)_{n^2})_{n\in\mathbb{N}} : \mathbb{Q}$$

  and this last term represents $\mathsf{approxFaster}$.

Since all terms-in-context represent measurable functions, $\mathsf{approxFaster}$ is measurable. This type of proof amounts to mechanically type-checking that the term and its sub-terms are well-formed. We call this proof technique *measurability by type-checking*.

Unfortunately, classical measure theory restricts this technique. Consider the function taking a measurable function $f : \mathbb{R} \to \mathbb{R}$ and returns a sequence of rational-valued approximations:

$$\mathsf{approxFun} : \mathbb{R}^{\mathbb{R}} \to (\mathbb{Q}^{\mathbb{R}})^{\mathbb{N}} \qquad \mathsf{approxFun}\, f \coloneqq (\lambda x.\mathsf{approx}(f\, x))_{n\in\mathbb{N}}$$

I.e., given a measurable function $f : \mathbb{R} \to \mathbb{R}$, we approximate it by a sequence of functions. Each function, given an argument $x : \mathbb{R}$, uses the measurable function approx to approximate $f\,x : \mathbb{R}$, the application of the given function f to the argument x. Measurability by type-checking needs:

- a measurable space structure $\mathbb{R}^{\mathbb{R}}$ over the set of measurable functions $f : \mathbb{R} \to \mathbb{R}$; and
- a term-former representing the application of a function-variable to an argument:

$$f : \mathbb{R}^{\mathbb{R}}, x : \mathbb{R} \vdash f\, x : \mathbb{R}$$

Aumann's theorem [1961] states that these two requirements are impossible.

In contrast, the quasi-Borel spaces of Staton et al. [2017] support spaces such as $\mathbb{R}^{\mathbb{R}}$ and term formers such as function application. If we consider terms-in-context as representing measurable functions between quasi-Borel spaces, then both measurability-by-type-checking proofs are well-posed and valid. Moreover, when dealing with concrete spaces such as $\mathbb{N}$, $\mathbb{Q}^{\mathbb{N}}$, $\mathbb{R}$, and more generally, any *standard Borel space*, measurability with respect to the quasi-Borel and measurable space structures agree. The situation is analogous to the conservativity of the extension of the real numbers by the complex numbers, who possess better closure properties. Even if we use quasi-Borel spaces to construct a measurable function between standard Borel spaces, conservativity implies this function is in fact measurable between them as measurable spaces.

Although by fiat quasi-Borel spaces enjoy a rich type structure, we need evidence this structure is meaningful for classical concepts in probability and statistics. To focus the discussion on concepts of pure measurability, we consider concrete measurable space structures for classes of well-behaved functions: the continuous functions and wider class of right-continuous with left-limits functions. These spaces are relevant in continuous-time process theory, and well-known results in the foundations of statistics show they have a well-behaved measurable space structure and support operations of interest such as function application as measurable functions. We will show that these two spaces embed as subspaces of the space of measurable functions $\mathbb{R}^{[a,b)}$.

Structure, contribution, and target audience

We aim for a relatively accessible account, targeting readers interested in discharging measurability proofs in structural ways. We therefore proceed at a relatively slow pace in the first 20 or so pages, focusing on exposition. We accompany each new concept by several examples, resulting in an unusual number of examples (over 80 examples). Readers who are familiar with quasi-Borel space theory specifically, or more broadly with semantic interpretations of simply-typed programming languages and the interplay between internal and external reasoning, may find this first part relatively standard and slow-paced, and may be able to quickly skim through sections §2–§4.

Contribution. The specific contributions we make over the existing literature are:

- Using coinduction to define measurable selection principles (Sec. 5).
- A measurable toolkit comprising of concepts and sufficient conditions for characterising the measurable events using a metric (Sec. 6).
- An embedding of the space of continuous functions into the function-space (Sec. 6)).
- A similar embedding of the Skorokhod space of càdlàg functions (Sec. 7)).

Moreover, the way we present these concepts adds to the existing literature. First, the quantity of examples we use is unparalleled. Second, we recount the theory of quasi-Borel spaces from first principles without relying on existing knowledge of measure theoretic results. We hope this structure will make this theory accessible to a wider audience.

We expect some readers may be interested in the more technical research context, both from the areas of categorical logic and type theory. As a compromise, we delimit technical segments setting up the research context as follows. Other readers can safely skip this handful of segments.

We delimit segments that relate to the broader area of categorical logic and categorical semantics with a 'beware cats' sign. For example, the fact that quasi-Borel spaces can interpret a simple type theory (i.e., form a bi-Cartesian-closed category) was known from the outset [Staton et al. 2017]. It follows by fiat from their construction as a Grothendieck quasi-topos. We therefore make no claim to the novelty of the type interpretation, only to its application to simply-typed measurability.

We delimit segments that relate to the broader area of type theory and logic, typically higher-order logic or simple-type theory, with the 'beware types' sign. For example, quasi-Borel spaces support reasoning in a higher-order logic (HOL) where two sub-object classifiers are in play: the propositions classify any subset; and the Booleans classify measurable events.

We include proofs or proof notes for most of our results. However, the proofs concerning the characterisation of the continuous and càdlàg function spaces as concrete and their generating metric are analytically involved. We have delegated them to an appendix. We encourage readers who want to employ simply-typed measurability proofs to skim these appendices.

2 BRIEF BACKGROUND

We will need some preliminaries from the theories of classical measurable spaces and quasi-Borel spaces. For the latter, we will keep the exposition to a minimum, and use simply-typed components to recapitulate and extend the theory. In both theories:

- a space consists of a set of points representing outcomes of a statistical model;
- we isolate subsets of outcomes representing events we wish to measure; and
- we isolate functions along which we can transfer measure from a well-behaved space.

The difference amounts to which of these notions we take as primitive.

2.1 Classical measure theory

Recall the following core analysis concepts covered by classical textbooks [Kechris 1995, e.g.] which we include for completeness. Most of this section can be skipped at first reading, only the definition of a σ-field, measurable space, measurable function, and the Borel subsets of the reals are needed for most of this manuscript.

2.1.1 Fields of subsets. Let X be a set. A *σ-field* $\mathcal{B}$ over X is a family of subsets $\mathcal{B} \subseteq \mathcal{P}\,X$, called *events*, satisfying the following closure properties:

- $\mathcal{B}$ is a Boolean sub-algebra of the powerset with its finitary set-theoretic operations, i.e., events are closed under finite unions and intersection, and complements:

$$\emptyset, X \in \mathcal{B} \qquad \frac{E, F \in \mathcal{B}}{E \cup F, E \cap F \in \mathcal{B}} \qquad \frac{\text{for all } i = 1, \ldots, n\text{: } E_i \in \mathcal{B}}{\bigcup_{i=1}^{n} E_i, \bigcap_{i=1}^{n} E_i \in \mathcal{B}} \qquad \frac{E \in \mathcal{B}}{E^{\complement} \in \mathcal{B}}$$

- events are closed under suprema of increasing chains:

$$\frac{E_{-} \in \mathcal{B}^{\mathbb{N}} \qquad \text{for all } n \in \mathbb{N}\text{: } E_n \subseteq E_{n+1}}{\bigcup_{n=1}^{\infty} E_n \in \mathcal{B}}$$

Example 1 (discrete, indiscrete, and generated σ-fields)**.** The powerset is a σ-field called the *discrete* σ-field over X, and $\{\emptyset, X\}$ is a σ-field called the *indiscrete* σ-field. For every family of subsets $\mathcal{A} \subseteq \mathcal{P}\,X$, the σ-field *generated* by $\mathcal{A}$ is the smallest σ-field $\sigma(\mathcal{A})$ over X containing $\mathcal{A}$. □

Example 2 (Borel subsets of the reals)**.** The class of *Borel subsets* of the real line $\mathcal{B}_{\mathbb{R}}$ is the σ-field generated by the open intervals, i.e. $\sigma(\{(a, b) | a, b \in \mathbb{R}\}$. We can equally well take as a generating set: the closed intervals $[a, b]$; the right-half-open intervals $[a, b)$, the open intervals with rational end-points, the lower-open intervals $(-\infty, a)$, etc. □

The literature sometimes uses the terminology: *σ-algebra* for σ-field, and *trivial* σ-field for the indiscrete σ-field. While axiomatisations for σ-fields vary, e.g., we may ask only for countable unions and complements, they result in the same class of structures.

Topological spaces are a central source for σ-fields for continuous sets of points. Recall that a *topology* over X is a family of subsets $\mathcal{O} \subseteq \mathcal{P}X$ closed under finite intersections and arbitrary unions. Metric spaces are a source for well-behaved topological space. Recall that a *metric* over X is a function $d : X^2 \to [0, \infty]$ that is:

- positive: $d(x, y) = 0 \iff x = y$, for all $x, y \in X$;
- symmetric: $d(x, y) = d(y, x)$, for all $x, y \in X$;
- satisfies the triangle inequality: $d(x, z) \leq d(x, y) + d(y, z)$, for all $x, y, z \in X$.

E.g., we define sequential limits by expressing the usual definition verbatim using the metric:

$$L = \lim_{n\to\infty} x_n \qquad \text{when} \qquad \forall\varepsilon > 0.\exists n \in \mathbb{N}.\forall m \geq n.d(L, x_m) < \varepsilon$$

Example 3 (discrete metric, metric topology, discrete topology, generated topology)**.** For every positive $a \in (0, \infty]$ constant, consider the function $\lambda\,(x, y)\,.\,\{x = y : 0; x \neq y : a\}$ which sends two points to 0 when they are equal and to a otherwise. It is always a metric we call a *discrete* metric. Every metric d generates a topology, the corresponding *metric* topology, given by:

$$U \in \mathcal{O}_d \iff \forall x \in U.\exists\varepsilon \in (0, \infty].\mathrm{B}^d_\varepsilon x \subseteq U \qquad \text{where} \qquad \mathrm{B}^d_\varepsilon x := \{y \in X | d(y, x) < \varepsilon\}$$

That is, a subset U is open if every point belongds to a 'disc' of positive radius ε that is contained in U. The *discrete* topology is the full powerset $\mathcal{P}X$, and it is the metric topology for every discrete metric on X. Given a family of subsets $\mathcal{A} \subseteq \mathcal{P}X$, the topology *generated* by $\mathcal{A}$ is the smallest topology $\mathcal{O}_{\mathcal{A}}$ containing $\mathcal{A}$. The metric topology is the topology generated by the discs of positive radius, i.e., by $\mathcal{A} := \left\{\mathrm{B}^d_\varepsilon x \middle| x \in X, \varepsilon \in (0, \infty]\right\}$. □

The purpose of these concepts is to introduce the concepts in this example:

Example 4 (Borel subsets of Euclidean, metric, and topological spaces)**.** The *Euclidean topology* on $\mathbb{R}$ is the topology generated by the *distance* metric $d(x, y) := |x - y|$. The Borel subsets of the reals is the σ-field generated by the Euclidean topology. Generalizing, the Borel subsets over a metric or a topology are the σ-field generated by the open subsets in the metric or given topology. □

2.1.2 Spaces and their functions. A *measurable space* A is a pair $(\llcorner A \lrcorner, \mathcal{B}_A)$ consisting of a set $\llcorner A \lrcorner$, whose elements we call *points*, and a σ-field $\mathcal{B}_A$ over these points. Similarly, a topological or metric space consists of a set of points equipped with a topology or a metric.

Example 5 (discrete, indiscrete, and Euclidean spaces)**.** We can turn every set X into the *discrete* measurable space $\ulcorner X \urcorner$ via $(X, \mathcal{P}X)$ and the *indiscrete* measurable space structure $[X]$ via $(X, \{\emptyset, X\})$. Each topological A provides a measurable space $\ulcorner A \urcorner$ via the Borel subsets $(\llcorner A \lrcorner, \sigma(\mathcal{O}_A))$. In particular, we will consider the real line $\mathbb{R}$ as a topological or measurable space, as needed, by equipping it with the Euclidean topology and its Borel σ-field. □

Let A, B be measurable spaces. A *(Borel) measurable function* $f : A \to B$ is a function between the sets of points $f : \llcorner A \lrcorner \to \llcorner B \lrcorner$ such that, for every event $E \in \mathcal{B}_B$, the inverse image is an event $f^{-1}[E] \in \mathcal{B}_A$. Simiarly, a *continuous function* $f : A \to B$ between topological spaces is a function $f : \llcorner A \lrcorner \to \llcorner B \lrcorner$ whose inverse image sends open subsets to open subsets.

Example 6 (measurability via continuity and composition)**.** Every continuous function between topological spaces is a measurable function between their corresponding measurable spaces. E.g., every constant function $\underline{a} := \lambda x.a$ is both continuous and measurable. The absolute value $|{-}| : \mathbb{R} \to \mathbb{R}$ is continuous and measurable. Identity functions is always continuous between the same topological space and always measurable between the same measurable space. Composing continuous or measurable functions preserves their continuity or measurability. □

Example 7 (measurable recombination)**.** Consider a countable sequence of measurable functions $(f_i : \llcorner A \lrcorner \to \llcorner B \lrcorner)_{i \in I}$, and a partition $\llcorner A \lrcorner = \biguplus_{i \in I} E_i$ comprising of events $(E_i)_{i \in I} \in \mathcal{B}_A^I$, i.e.: $\llcorner A \lrcorner = \bigcup_{i \in I} E_i$ and for $i \neq j$, we have $E_i \cap E_j = \emptyset$. The corresponding *recombination* function $[E_i.f_i]_{i \in I} : \llcorner A \lrcorner \to \llcorner B \lrcorner$ sends each $r \in E_i$ to $f_i\, r$. If each component function is measurable $f_i : A \to B$, then their recombination is a measurable function $[E_i.f_i]_{i \in I} : \llcorner A \lrcorner \to \llcorner B \lrcorner$. Recombination functions are not typically continuous. For example, given an event $E \in \mathcal{B}_A$, the following *characteristic* and *indicator* functions are measurable:

$$(- \in E) : A \to \ulcorner \{\mathbf{true}, \mathbf{false}\} \urcorner =: \mathbb{B} \qquad [- \in E] : A \to \mathbb{R}$$
$$(x \in E) := \left[E.\mathbf{true}, E^{\complement}.\mathbf{false}\right] \qquad [x \in E] := \left[E.0, E^{\complement}.1\right]$$

However, the characteristic function is not a continuous function $(- \in (0,1)) : \mathbb{R} \to \mathbb{B}$. The characteristic functions form a bijective correspondence between Borel subsets and measurable functions from A to $\mathbb{B}$. □

A *Borel isomorphism* $f : A \xrightarrow{\cong} B$ is a bijective measurable function between the two measurable spaces, whose inverse is a measurable function between them. The corresponding topological notion is called a *homeomorphism.*

Example 8 (subspace embedding)**.** Let A be a measurable space and consider a subset of points $X \subseteq \llcorner A \lrcorner$. We equip this subset with the *subspace* σ-field $\mathcal{B}_X := \{E \cap X | E \in \mathcal{B}_A\}$, consisting of the events in A restricted to the points in X. The inclusion is then a measurable function $\lambda x.x : X \to A$. We can thus talk about subsets such as $(0,1) \subseteq \mathbb{R}$ as measurable space of their own right. More generally, a *subspace embedding* $f : A \hookrightarrow B$ is a measurable function whose restriction to the image subspace is a Borel isomorphism $\lambda x.f\,x : A \xrightarrow{\cong} f[A]$. □

Example 9 (random elements)**.** A *random element* in a measurable space A is a measurable function $\alpha : \mathbb{R} \to A$. By Examples 6–7, the constant elements include the constant functions, and they are closed under precomposition with a Borel measurable function $\varphi : \mathbb{R} \to \mathbb{R}$ and under countable recombination. □

2.1.3 Standard Borel spaces. Classical measure theory works best when we apply it to Borel subsets of the real line. A measurable space A is a *standard Borel space* when there is a Borel subset $E \in \mathcal{B}_{\mathbb{R}}$ and a Borel isomorphism $f : A \xrightarrow{\cong} E$. A cornerstone result in classical descriptive set theory relates the measurable spaces that are standard Borel to the so called *Polish* topological spaces. To define them, recall that a Cauchy sequence $(x_n)_{n \in \mathbb{N}}$ in a metric space is a sequence whose suffixes grow infinitesimally closer:

$$\forall \varepsilon > 0. \exists n \in \mathbb{N}. \forall m, k \geq n. d(x_m, x_k) < \varepsilon.$$

A metric is *Cauchy-complete* when every Cauchy sequence has a limit. Thus a topological space is *completely metrisable* when its topology is the metric topology for a Cauchy-complete metric. A topological space is *separable* when there is a countable sequence of open sets $(U_i)_{i \in I} \in \mathcal{O}$ such that every open set V can be represented as $V = \bigcup_{i \in I, U_i \subseteq V} U_i$. A topological space is *Polish* when it is separable and completely metrisable.

The celebrated characterisation of standard Borel spaces is then:

Theorem 1. *Let A be a measurable space. The following are equivalent:*

- *A is standard Borel.*
- *A is Borel isomorphic to a discrete countable space or to $\mathbb{R}$.*
- *A is the measurable space induced by some Polish space.*

The standard Borel spaces enjoy useful closure properties: they are closed under countable products and coproducts, and consequently are closed under exponentiating by a discrete space. In detail, given an I-ary collection of measurable spaces $(A_i)_{i\in I}$, their product and coproducts are given by the Cartesian product and disjoint union:

$$\llcorner \textstyle\prod_{i\in I} A_i \lrcorner := \prod_{i\in I} \llcorner A_i \lrcorner \qquad \mathcal{B}_{\prod_{i\in I} A_i} := \sigma\left\{ \prod_{i\in I} E_i \in \prod_{i\in I} \mathcal{B}_{A_i} \,\middle|\, \begin{array}{l}\text{for all but countably many } i\text{:}\\ E_i = \llcorner A_i \lrcorner\end{array} \right\}$$

$$\llcorner \textstyle\coprod_{i\in I} A_i \lrcorner := \coprod\limits_{i\in I} \llcorner A_i \lrcorner := \bigcup\limits_{i\in I} \{i\} \times \llcorner A_i \lrcorner \qquad \mathcal{B}_{\coprod_{i\in I} A_i} := \left\{ \bigcup\limits_{i\in I} \{i\} \times E_i \subseteq \coprod_{i\in I} \llcorner A_i \lrcorner \,\middle|\, \forall i \in I. E_i \in \mathcal{B}_{A_i} \right\}$$

$$\pi_j : \textstyle\prod_{i\in I} A_i \to A_j \qquad \pi_j\,(x_i)_{i\in I} := x_j \qquad \iota_j : A_j \to \coprod_{i\in I} A_i \qquad \iota_j x := (j, x)$$

Given measurable spaces A and B, let $\mathbf{Meas}(A, B)$ be the set of measurable functions between them. An *exponential* of B by a measurable space A amounts to a σ-field$\mathcal{B}_{B^A}$ over B such that, letting B^A be the measurable space $(\mathbf{Meas}(A, B), \mathcal{B}_{B^A})$:

- the evaluation function given by $\mathsf{eval}(f, x) := f\,x$ is measurable: $\mathsf{eval} : B^A \times A \to B$
- for all measurable spaces C, a function $f : C \times A \to B$ is measurable iff the curried function $\lambda z.\lambda x.f(z, x) : C \to B^A$ is measurable.

Via the correspondence between events and characteristic functions, an exponential of $\mathbb{B}$ by A implies there is σ-field $\mathcal{B}_{\mathcal{B}_A}$ for $\mathcal{B}_A$ where set membership is measurable: $(\in) : A \times \mathcal{B}_A \to \mathbb{B}$.

Theorem 2 (Aumann [1961])**.** *The exponentials $\mathbb{R}^{\mathbb{R}}$ and $\mathbb{B}^{\mathbb{R}}$ do not exist, as neither evaluation nor set membership are measurable for all σ-fields over* $\mathbf{Meas}(\mathbb{R}, \mathbb{R})$ *or* $\mathcal{B}_{\mathbb{R}}$.

Example 10 (restricted simply-typed measurability)**.** Despite Aumann's theorem, we have the exponential $\mathbb{Q}^{\mathbb{N}} := \prod_{n\in\mathbb{N}} \mathbb{Q}$. Using some enumeration of the rationals $\lambda n.q_n : \mathbb{N} \xrightarrow{\cong} \mathbb{Q}$, define:

$$\begin{array}{l}\mathsf{approx},\\ \mathsf{approxFaster}\end{array} : \mathbb{R} \to \mathbb{Q}^{\mathbb{N}} \qquad \mathsf{approx}\,r := \lambda n.q_{\min\left\{k \,\middle|\, q_k \in (r - \frac{1}{n}, r)\right\}} \qquad \mathsf{approxFaster}\,r := \mathsf{eval}(\mathsf{approx}\,r, n^2)$$

We prove that $\mathsf{approxFaster}$ is measurable by composing measurable functions. For approx, define:

$$\mathcal{B}_{\mathbb{R}} \ni F_{n,k} := (q_k, q_k + \tfrac{1}{n}) \cap \bigcap_{i=1}^{k-1} (q_k, q_k + \tfrac{1}{n})^{\complement} \qquad \text{N.B.:} \qquad r \in F_{n,k} \iff (\mathsf{approx}\,r)_n = k$$

Therefore, $\lambda r.\lambda n.(\mathsf{approx}\,r)_n : \mathbb{R} \times \mathbb{N} \to \mathbb{Q}$ is measurable, and so approx is also measurable. □

2.2 Quasi-Borel spaces

Classical measurre theory axiomatises spaces via their measurable events, and derives the admissible random elements (Example 9), the subspace embeddings, and the induced and well-behaved standard Borel spaces. The theory of quasi-Borel spaces of Staton et al. [2017] pivots these concepts, and axiomatises spaces via their admissible random elements and derives the remainder. It builds over the well-behaved fragment of measurable spaces, namely the σ-field of Borel sets (cf. Example 4), i.e., the smallest family of $\mathbb{R}$-subsets containing the intervals and closed under complements and countable unions and intersections, and the concept of a Borel measurable function $f : \mathbb{R} \to \mathbb{R}$ (cf. §2.1.2), i.e., a function whose inverse image sends a Borel set to a Borel set.

Formally, a *metaphorology*[1] over a set X is a family of functions $\mathcal{R} \subseteq X^{\mathbb{R}}$, called *random elements*, satisfying the following closure properties:

- Every constant function is a random element: $\underline{a} := \lambda r.a \in \mathcal{R}$.

- Closure under precomposition by a Borel measurable function:

$$\frac{\varphi : \mathbb{R} \longrightarrow \mathbb{R} \qquad \alpha \in \mathcal{R}}{\alpha \circ \varphi \in \mathcal{R}}$$

- Closure under countable recombination (cf. Example 7):

$$\frac{I \text{ countable} \qquad (E_i)_{i\in I} \in \mathcal{B}_{\mathbb{R}} \qquad \mathbb{R} = \biguplus_{i\in I} E_i \qquad (\alpha_i)_{i\in I} \in \mathcal{R}^I}{[E_i.\alpha_i]_{i\in I} := \lambda \{r \in E_i : \alpha_i r | i \in I\} \in \mathcal{R}}$$

A *quasi-Borel space* X is a pair $(\llcorner X \lrcorner, \mathcal{R}_X)$ consisting of a set $\llcorner X \lrcorner$, whose elements we call *points*, and a metaphorology $\mathcal{R}_X$ over it. Henceforth, to avoid the awkward initialism 'qbs', when we say 'space' we mean a quasi-Borel space.

Example 11 (real line, measurable spaces as quasi-Borel spaces)**.** Using Example 9, the real line equipped with its random elements $\mathcal{R}_{\mathbb{R}} := \mathbf{Meas}(\mathbb{R}, \mathbb{R})$ is a space. Generalising, the random elements of a measurable space A make it into a space: $\llcorner A \lrcorner_{\mathbf{Qbs}} := (\llcorner A \lrcorner, \mathbf{Meas}(\mathbb{R}, A))$. □

Example 12 (σ-simple functions, discrete and indiscrete spaces)**.** Let X be a set. A *σ-simple function* $\alpha : \mathbb{R} \longrightarrow X$ is a countable recombination of constant functions $\alpha = \left[E_i.\underline{x_i}\right]_{i\in I}$ where I is countable, $(E_i)_{i\in I} \in \mathcal{B}^I_{\mathbb{R}}$ is a partition $\mathbb{R} = \biguplus_{i\in I} E_i$ and $(x_i)_{i\in I} \in X^I$ is an I-indexed family of constants. The σ-simple functions in X form a metaphorology $\mathcal{R}_{\ulcorner X \urcorner}$. We define the *discrete* space over X as $\ulcorner X \urcorner := \left(X, \mathcal{R}_{\ulcorner X \urcorner}\right)$. The collection of all functions $\alpha : \mathbb{R} \longrightarrow X$ is also a metaphorology $\mathcal{R}_{[X]}$. We define the *indiscrete* space over X as $[X] := \left(X, \mathcal{R}_{[X]}\right)$. The discrete spaces let us embed set theoretic constructions 'as is'. As we will see later, the indiscrete spaces will let us embed set-theoretic reasoning into measurability constructions. □

Let X, Y be spaces. A (quasi-measurable) function $f : X \longrightarrow Y$ is a function $f : \llcorner X \lrcorner \longrightarrow \llcorner Y \lrcorner$ between their sets of points that preserves random elements:

$$\forall \alpha \in \mathcal{R}_X . f \circ \alpha \in \mathcal{R}_Y$$

Example 13 (random elements are measurable)**.** Every random element $\alpha \in \mathcal{R}_X$ in a space X is a measurable function $\alpha : \mathbb{R} \longrightarrow X$ due to the precomposition axiom. Conversely, a measurable function $\alpha : \mathbb{R} \longrightarrow X$ must also be a random element since $\alpha = \alpha \circ \mathrm{id}$ and the identity function $\mathrm{id} : \mathbb{R} \longrightarrow \mathbb{R}$ is a random element in $\mathbb{R}$. □

Example 14 (measurable functions are quasi-measurable)**.** Given measurable spaces A and B, every measurable function $f : A \longrightarrow B$ is also (quasi-)measurable between the underlying spaces $f : \llcorner A \lrcorner_{\mathbf{Qbs}} \longrightarrow \llcorner B \lrcorner_{\mathbf{Qbs}}$. The converse may fail (cf. Sec. **??**)—we have more quasi-measurable functions (but cf. Example 25). Through this example we now inherited many examples for quasi-measurable functions between spaces of interest. □

Example 15 (identities and composition)**.** Every identity function $\mathrm{id} : X \longrightarrow X$ is measurable over the space, and composing measurable functions yields a measurable function. □

We will exhibit many more examples as we introduce our typed development.

[1] Going back to its original roots, 'metaphor' originates from the Greek $\mu\varepsilon\tau\alpha$ ('meta', across) and $\varphi\varepsilon\rho\omega$ ('phero', to carry). This choice makes 'metaphors' an appealing alternative terminology to 'random element'.

Type	$\ni A, B ::=$	types		$\mid \dashv\!\mid X \mid\!\vdash$	semantic type reflection
	$A \to B$	function type		$\mid \cdots$	
preTerm	$\ni M, N ::=$		terms		
	x	variable		$\mid \mathbf{let}\, x = M \;\mathbf{in}\; N$	intermediate result
$\mid$	$\lambda x : A.M$	abstraction		$\mid \dashv\!\mid f : G \to X \mid\!\vdash_A$	semantic reflection
$\mid$	$M\, N$	application		$\mid \cdots$	

Fig. 1. Core constructs

3 SIMPLE TYPE THEORY

Over the next few sections we will develop a simple programming language for expressing measurable functions. We will introduce the language gradually for simplicity. For convenience, we summarise the full language in Appendix A. It is a variation on the calculi introduced by [Dash et al. 2023; Ścibior et al. 2018]

3.1 Base constructs

To introduce the core constructs of the language, we demonstrate a simple measurability proof. Consider continuous function $\lambda x.\frac{1}{x} : (0, 1] \to [1, \infty)$. Its continuity amounts to this property:

$$\forall x \in (0, 1].\forall \varepsilon > 0.\exists \delta > 0.\forall y \in (0, 1].\, |x - y| < \delta \implies \left|\frac{1}{x} - \frac{1}{y}\right| < \varepsilon$$

We use our language to prove that the dependency of δ on x and ε can be made measurable. Out of a typical proof for continuity (below, on the right), we define the function on the left:

$$\begin{aligned}
&\delta : (0, 1] \times (0, \infty) \to (0, \infty)\\
&\delta := \lambda x : (0, 1], \varepsilon : (0, \infty).\\
&\quad \mathbf{let}\ \delta_1 = \tfrac{1}{2}x\\
&\qquad\ \ \delta_2 = \tfrac{\varepsilon x^2}{2}\ \mathbf{in}\ \min(\delta_1, \delta_2)
\end{aligned}$$

Proof ($\lambda x.\frac{1}{x}$ is continuous)
Take $x \in (0, 1]$ and $\varepsilon > 0$. Note $\delta(x, \varepsilon) \le \frac{1}{2}x$. If $|x - y| \le \delta(x, \varepsilon)$:
$y \ge \frac{1}{2}x \Rightarrow \frac{1}{xy} \le \frac{2}{x^2} \Rightarrow \left|\frac{1}{x} - \frac{1}{y}\right| = \frac{1}{xy}\,|y - x| < \frac{2}{x^2} \cdot \frac{\varepsilon x^2}{2} = \varepsilon.$ ∎

The program on the left abstracts over the variables x and ε. It then introduces two intermediate variables δ_1 and δ_2 that play a role in the proof, and then uses them to calculate the final result.

Typically, we would reason: since all intermediate functions are continuous, this function is measurable, even continuous. Here, however, the reasoning is more nuanced: since all the primitive functions— $\lambda x.\frac{1}{2}x$, $\lambda x, \varepsilon.\frac{\varepsilon x^2}{2}$, and min—are measurable, we can embed them into the program that calculates $\delta(x, \varepsilon)$. Since we expressed δ with a program, it is measurable. Refactoring the typical argument to go through a program is a detour in this simple case. We will later consider more sophisticated examples of measurable constructions. In those examples, the clear recourse to a measurable formalism makes it straightforward to ensure a construction is measurable.

Summarising, the basic constructs in our calculus include:

- introducing function variables and intermediate variables;
- embedding measurable functions into a program, drawing variables from their context.

Fig. 1 presents the basic constructs in our proposed simple-type system. We present additional type formers in §3.4–§**??**. Our types include function types $A \to B$ and we include spaces as types by a *type reflection* construct. For example: the reals $\dashv\!\mid \mathbb{R} \mid\!\vdash$, the positive reals $\dashv\!\mid (0, \infty) \mid\!\vdash$, and the half-open interval $\dashv\!\mid (0, 1] \mid\!\vdash$. We suppress this type reflection construct in examples. We will carve the well formed terms from the set of pre-terms via the type-system in Fig. 2. Pre-terms include the standard variables, function abstraction ($\lambda x : A.M$)—binding the variable x in the body M—and application constructs, and a let-binding construct ($\mathbf{let}\, x = M \;\mathbf{in}\; N$) for binding the variable x to the intermediate results of M in the body N. The semantic reflection construct is non-standard. It

$$\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A} \qquad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : B} \qquad \frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\,N : B}$$

$$\frac{\Gamma \vdash M : A \qquad \Gamma, x : A \vdash N : B}{\Gamma \vdash \mathbf{let}\, x = M \;\mathbf{in}\; N : B} \qquad \frac{\Gamma \text{ denotes } G \qquad A \text{ denotes } X \qquad f : G \to X \text{ measurable}}{\Gamma \vdash \dashv\!\mid f : G \to X \mid\!\vdash_A : A}$$

Fig. 2. Simple-type system: rules for the core constructs

allows us to incorporate measurable functions between appropriate spaces into our programs. For example, $\dashv\!\mid \lambda u, v.\frac{uv^2}{2} \mid\!\vdash_{[0,1)\to(0,\infty)\to(0,\infty)}$. We suppress this construct in examples.

⚠ Typically one expects the syntax to form a relatively small set. Incorporating concrete spaces through semantic type reflection and arbitrary functions through semantic reflection may seem at odds with this convention. For our purposes, it suffices to define a family of languages parameterised by a countable collection of primitive type symbols and built-in constants. We can think of the semantic reflection constructs as an anonymous definition for a primitive type or built-in constant. More formally, in our running example for the the $\delta(x, \varepsilon)$ dependency function, we instantiate with 2 primitive types: $(0, 1]$ and $(0, \infty)$; and 3 constants: $\lambda x.\frac{1}{2}x$, $\lambda x, \varepsilon.\frac{\varepsilon x^2}{2}$, and min.

We define the type system as a 3 place well-formedness relation $\Gamma \vdash M : A$ relating a pre-term M with a type A under a a *typing context* Γ in the set $\mathbf{Ctxt} := \mathbf{Var} \rightharpoonup_{\text{fin}} \mathbf{Type}$ of finitely-supported maps of variable names to types. We write $(x : A) \in \Gamma$ when x is in the domain of Γ and Γ assigns it the type A. We require the domain of Γ to include the free variables of M. The rules, in Fig. 2, are mostly standard: variables are typed by their assigned type; function abstraction types the body in a context extended with a fresh variable binding the abstracted variable; application requires the function to type as a function; and let-binding types the body in a context extended with the type of the intermediate program. The non-standard rule is for the non-standard semantic-reflection construct. It is tied to the spaces the typing context and the result type denote. The rule requires a calculation, external to the type-system, of these spaces, and an external proof that the reflected function is indeed measurable between these spaces. The *closed* terms of type A are the terms that are well formed in the empty context $\vdash M : A$.

⚠ The typing rule for reflection couples the type-system with the denotational semantics we define in the next section. Since we require type annotation on every semantic reflection, this requirement does not preclude algorithmic type-checking. While this design is non-standard, the result of such a hypothetical type-checker is a list of remaining proof obligations for discharging occurrences of semantic reflection rule. This kind of design is fairly common in proof assistants, where the programmer can leave holes or postulates for the type-checker.

3.2 Product and function spaces

The sole reason for considering this language is to use it to define spaces and measurable functions between them. In order to relate the core constructs—contexts, function types, and reflection—we need two constructions: the product and function space.

Given two spaces X, Y, their product space $X \times Y$ consists of the following metaphorology over the product of their points, equipped with the standard component projections:

$$\llcorner X \times Y \lrcorner := \llcorner X \lrcorner \times \llcorner Y \lrcorner \qquad \mathcal{R}_{X\times Y} := \{\lambda r.\,(\alpha\, r, \beta\, r) | \alpha \in X, \beta \in Y\} \qquad X \xleftarrow{\pi_1} X \times Y \xrightarrow{\pi_2} Y$$

One can motivate this definition by thinking of $\lambda r.\,(\alpha\, r, \beta\, r)$ as two random variables that are correlated by the random sample r. More generally, the product $\prod_{i\in I} X_i$ of an I-indexed family of

$$[\![A \to B]\!] := [\![B]\!]^{[\![A]\!]} \qquad [\![\dashv\!|X|\!\vdash]\!] := X \qquad [\![\Gamma]\!] := \prod_{(x:A)\in\Gamma} [\![A]\!]$$

$$[\![x]\!]\,\gamma := \pi_x \gamma \qquad [\![\lambda x : A.M]\!]\,\gamma := \lambda u.\,[\![M]\!]\,(\gamma, u) \qquad [\![M\,N]\!]\,\gamma := \mathsf{eval}\,([\![M]\!]\,\gamma, [\![N]\!]\,\gamma)$$

$$[\![\mathbf{let}\, x = M \,\mathbf{in}\, N]\!]\,\gamma := [\![N]\!]\,(\gamma, [\![M]\!]\,\gamma) \qquad [\![\dashv\!|f : [\![\Gamma]\!] \to [\![A]\!]|\!\vdash_A]\!]\,\gamma := f\,\gamma$$

Fig. 3. Denotations of (a, top) core simple types (b, bottom) core constructs

spaces $(X_i)_{i\in I}$ consists of the following metaphorology over the product of their points:

$$\llcorner \prod_{i\in I} X_i \lrcorner := \prod_{i\in I} \llcorner X_i \lrcorner \qquad \mathcal{R}_{\prod_i X_i} := \left\{\lambda r.\, (\alpha_i r)_{i\in I} \middle| \forall i.\alpha_i \in \mathcal{R}_{X_i}\right\} \qquad \prod_{i\in I} X_i \xrightarrow{\pi_j} X_j$$

This notion of product is compatible with the product of measurable spaces. Following on from Example 14, given a family $(A_i)_{i\in I}$ of measurable spaces, the space $\llcorner \prod_{i\in I} A_i \lrcorner_{\mathbf{Qbs}}$ is the product $\prod_{i\in I} \llcorner A_i \lrcorner_{\mathbf{Qbs}}$. Functions such as $\min : (0,\infty)\times(0,\infty) \to (0,\infty)$, that are measurable in multiple variables, are therefore also quasi-measurable functions from the product of the underlying spaces.

The function space Y^X consists of the following metaphorology over the set of quasi-measurable functions $\mathbf{Qbs}(X, Y)$, equipped with the standard evaluation function:

$$\llcorner Y^X \lrcorner := \mathbf{Qbs}(X, Y) \qquad \mathcal{R}_{Y^X} := \{\lambda r.\lambda x.f(r, x) | f : \mathbb{R}\times X \to Y\} \qquad \mathsf{eval} : Y^X \times X \xrightarrow{\lambda(f,x).f\,x} Y$$

As usual, there is a natural bijection between measurable functions $f : G\times X \to Y$ and measurable functions $g : G \to Y^X$ given by $g = \lambda\gamma.\lambda x.f(\gamma, x)$ and $f(\gamma, x) = g\,\gamma\,x$. We will explore function spaces in great depth in the next section.

3.3 Denotational semantics

We can now define the relationship between syntactic types and the spaces they denote. Fig. 3(a) summarises these: function types denote function spaces; reflected spaces denote themselves; and contexts denote the product of their constituent types.

A term $\Gamma \vdash M : A$ denotes a measurable function $[\![M]\!] : [\![\Gamma]\!] \to [\![A]\!]$, following the standard interpretation in Fig. 3(b). Given an environment $\gamma \in [\![\Gamma]\!]$ for the free variables: variables denote their component in this environment; syntactic abstraction denotes the function that extends the semantics of the body with the input value; syntactic application denotes evaluation of the argument function at the argument; and let-binding extends the environment by the semantics of the intermediate sub-term. Reflection, noting the well-formedness condition, denotes itself.

⚠ The non-standard reflection construct is fairly conservative nonetheless. There is a long tradition in the categorical semantics of internal languages concerning the interplay between internal entities like types and terms, and externally constructed morphisms in the semantic domain of interest. All we have done here is set-up a family of languages that makes the boundary between an internal and the external language more flexible. Doing so comes at a price: we are not working with one language, but with a family of languages, potentially one for every function we wish to exhibit as measurable.

We have now fully described the core constructs in the language. Going back to the motivating example, we can now note that the proof reasons about dependency of δ on x and ε given by the

$$
\begin{array}{llll}
\mathbf{Type} \ni A, B ::= \cdots & \text{types} & & \text{type denotations:} \\
\quad \mid \ \{\rho\} & & \text{variant type } (\mathrm{Dom}\,\rho \text{ countable}) & [\![\{\rho\}\ ]\!] := \coprod_{(\ell:A)\in\rho} [\![A]\!] \\
\quad \mid \ \lhd\rho\rhd & & \text{record type } (\mathrm{Dom}\,\rho \text{ finite}) & [\![\lhd\rho\rhd]\!] := \prod_{(\ell:A)\in\rho} [\![A]\!]
\end{array}
$$

$$
\begin{array}{lll}
\mathbf{preTerm} \ni M, N ::= & \cdots & \text{terms} \\
\quad \mid \ A.\ell\, M & & \text{data constructor} \\
\quad \mid \ \mathbf{case}\, M\, \mathbf{of}\, \{\ell_\ell x_\ell.M_\ell \mid \ell \in I\} & & \text{pattern match} \\
\quad \mid \ \mathbf{<}\ell_1 : M_1, \ldots, \ell_n : M_n\mathbf{>} & & \text{record} \\
\quad \mid \ \mathbf{case}\, M\, \mathbf{of}\, \mathbf{<}\ell : x_\ell \mid \ell \in I\mathbf{>}.N & & \text{pattern match}
\end{array}
$$

$$
\frac{(\ell : A) \in \rho \qquad \Gamma \vdash M : A}{\Gamma \vdash \{\rho\}.\ell\, M : \{\rho\}}
\qquad
\frac{\Gamma \vdash M : \{\rho\} \qquad \text{for all } (\ell : A) \in \rho\text{: } \Gamma, x_\ell : A_\ell \vdash M :}{\mathbf{case}\, M\, \mathbf{of}\, \{\ell_\ell x_\ell.M_\ell \mid \ell \in \mathrm{Dom}\,\rho\}}
$$

$$
\frac{\text{for all } \ell \in I\text{: } \Gamma \vdash M_\ell : A_\ell}{\Gamma \vdash \mathbf{<}\ell := M_\ell \mid \ell \in I\mathbf{>} : \lhd\ell : A_\ell \mid \ell \in I\rhd}
\qquad
\frac{\Gamma \vdash M : \lhd\ell : A_\ell \mid \ell \in I\rhd \qquad \Gamma, (x_\ell : A_\ell)_{\ell\in I} \vdash N}{\Gamma \vdash \mathbf{case}\, M\, \mathbf{of}\, \mathbf{<}\ell : x_\ell \mid \ell \in I\mathbf{>}.N : B}
$$

Fig. 4. (a, top left) Variant and record types, and their: denotation (b, top right); (c, middle) terms and (d, bottom) typing rules

semantics of the closed term $[\![\delta]\!]\,() : (0,1] \times (0,\infty) \to (0,\infty)$. The remainder of this section extends the core with useful language constructs: variants, records, and inductive types.

3.4 Variants, records, and exhaustive pattern-matching

A *row* ρ is a set $\mathrm{Dom}\rho$, whose elements we call *labels*, together with mapping $\rho : \mathrm{Dom}\rho \to \mathbf{Type}$. We will typically use syntactically distinct elements for the labels in $\mathrm{Dom}\rho$, such as bold-face labels for concrete labels and ℓ for label metavariables. We will use semi-colon separated lists to describe concrete rows to help distinguish them from typing contexts.

Rows describe the shape of both *variant* types: tagged/disjoint unions of their constituent types; and *record* types: tuples of labelled fields. Fig. 4(a) presents the syntax for variant and record types, and subfig. (c) and (d) present their associated terms and type system.

Variants data constructors ($A.\ell M$) tag the result with the label, and variant pattern-matches ($\mathbf{case}\, M\, \mathbf{of}\, \{\ell_\ell x_\ell.M_\ell \mid \ell \in I\}$) allow us to discriminate based on this label, binding the tagged value to x_ℓ on the branch M_ℓ for each label ℓ. The abstract syntax does not discriminate the order of the branches. For simplicity, we only support variants with countably many labels, but it is straightforward to relax this restriction.

Records ($\mathbf{<}\ell_1 : M_1, \ldots, \ell_n : M_n\mathbf{>}$ tuple together labelled-fields. Here the abstract syntax does discriminate the order of the fields. The tuples they do not do not discriminate the order, just the content of the fields. Record pattern-matches ($\mathbf{case}\, M\, \mathbf{of}\, \mathbf{<}\ell : x_\ell \mid \ell \in I\mathbf{>}.N$) bring into scope all the fields of the record M. We only allow records of finite rows, as otherwise matching on a record would make our contexts infinite. While different designs may avoid this issue, e.g., match on a finite sub-row of the record instead, keeping to finite record rows suffices.

Example 16 (empty and unit types)**.** The empty type is the variant type for the empty row $\{\}$. The unit type is the record type for the empty row $\lhd\rhd$, inhabited by the unit record $\mathbf{<>} : \lhd\rhd$. □

Syntactic sugar. When we write _ in a binding occurrence, we desugar it into a fresh variable that we will not use in its scope. When we do not specify the type of a row label, we will desugar it into the unit type $\lhd\rhd$, and call it a *pure label*. For variant types, we will omit the unit record when using the data constructor corresponding to a pure label, desugaring ℓ to $\ell\mathbf{<>}$ in term positions and to

$\ell\_$ in binding occurrences in pattern-matches. Given a record $M : \triangleleft\rho\triangleright$ and a field $(\ell : A) \in \rho$, we elaborate the field projection $M.\ell : A_\ell$ into the record pattern-match $\mathbf{case}\, M\, \mathbf{of}\ \mathbf{<}\ell' x_{\ell'} | \ell' \in \rho\mathbf{>}.x_\ell$.

Example 17 (Booleans, enumerations, indices)**.** The Boolean type $\mathbb{B}$ is the variant type $\{\mathbf{true}, \mathbf{false}\}$, inhabited by **true** and **false**. The Boolean pattern-matching $\mathbf{case}\, M\, \mathbf{of}\ \{\mathbf{true}.N; \mathbf{false}.K\}$ is often called a *conditional* and uses the syntax $\mathbf{if}\, M\, \mathbf{then}\, N\, \mathbf{else}\, K$. More generally, an *enum* type is a variant type for a row of pure labels $\{\ell_1, \dots, \ell_n\}$. For example, given a natural number n, the type of n-indices is the enum $\mathbf{Fin}\, n \coloneqq \{0, 1, \dots, n-1\}$ for the row of labels representing the natural numbers below n. In particular, the bits are the 2-indices $\mathbf{Bit} \coloneqq \mathbf{Fin}\, 2$. □

Example 18 (tuples, immutable arrays, fixed-length numbers, tensors)**.** Given a sequence of types $A_1, \dots, A_n$, the *product type* $A_1 \times \cdots \times A_n$ is the record type $\triangleleft 1 : A_1, \dots, n : A_n\triangleright$. We desugar the tuple notation $(M_1, \dots, M_n)$ into the record $\mathbf{<}1 : M_1, \dots, n : M_n\mathbf{>}$. The type $A[n]$ of *immutable arrays* of type A and length n is the function type $\mathbf{Fin}\, n \to A$. The *fixed-length numbers* are the immutable arrays of bits, e.g. $\mathbf{Bit}[64]$. The type of *tensors* of shape ρ is the function type $\triangleleft\rho\triangleright \to \mathbb{R}$. The types $\mathbf{Tensor}\,[n_1, \dots, n_k]$ of k-dimensional tensors and the type $\mathbf{Index}\,[n_1, \dots, n_k]$ of indices of of shape $n_1, \dots, n_k$ are:

$$\mathbf{Index}\,[n_1, \dots, n_k] \coloneqq \mathbf{Fin}\, n_1 \times \cdots \times \mathbf{Fin}\, n_k \qquad \mathbf{Tensor}\,[n_1, \dots, n_k] \coloneqq \mathbf{Index}\,[n_1, \dots, n_k] \to \mathbb{R}$$ □

Example 19 (parametric distributions)**.** Distributions of practical interests are often fully specified by a simple type of parameters. For example, the type of Bernoulli distributions is the type of bias parameters $\mathbf{Bernoulli} \coloneqq \triangleleft\mathsf{bias} : [0, 1]\triangleright$; the type of 1-dimensional normal distributions is the type of their mean and standard deviation parameters $\mathbf{Gaussian}^1 \coloneqq \triangleleft\mathsf{mean} : \mathbb{R}, \mathsf{sdv} : (0, \infty)\triangleright$; etc. □

Example 20. The *hereditarily finite* types are the smallest class of types closed under function types between, and variants and records of finite rows of, themselves. All the examples so far are hereditarily finite types when they involve finite rows of hereditarily finite types. We will not make essential use of the hereditarily finite types here. □

We hope this long list of examples demonstrates the versatility of these types. For example:

Nested pattern-matching. We will use a compact syntax for exhaustive pattern-matching, collapsing several nested matches into a single match with nested patterns. For example, we elaborate:

$$\mathbf{case}\, M\, \mathbf{of}\ \{(\mathbf{true}, n)\,.M_1; (\mathbf{false}, n).M_2\} \qquad \text{to} \qquad \mathbf{case}\, M\, \mathbf{of}\ (b, n)\,.\mathbf{case}\, b\, \mathbf{of}\ \{\mathbf{true}.M_1; \mathbf{false}, M_2\}$$

We will shorten function abstractions that immediately match on their argument, and omit the annotation on the argument when it can be inferred, e.g.: $\lambda\,(i, j)\,.(i + j) : \mathbf{Tensor}\,[3, 4]$.

Denotations. Fig. 4(b) presents the denotation of variant and record types. Record types denote products of their constituent denotation, which we covered in the previous §3.3. Variant types denote the coproduct/disjoint union of spaces. Formally, given a family of spaces $(X_i)_{i \in I}$, the coproduct space has as points the disjoint union of their points, i.e. $\llcorner \coprod_{i \in I} X_i \lrcorner \coloneqq \coprod_{i \in I} \llcorner X_i \lrcorner$ the following metaphorology, given by recombination of random elements, makes the coproduct injections $\iota_j : X_j \to \coprod_i X_i$ measurable:

$$\mathcal{R}_{\coprod_{i \in I} X_i} \coloneqq \left\{ \left[E_j.\alpha_j\right]_{j \in J} \middle| J \subseteq I \text{ countable}, \left(E_j\right)_j \in \mathcal{B}_{\mathbb{R}}^J, \mathbb{R} = \biguplus_{j \in J} E_j, \left(\alpha_j\right)_j \in \prod_{j \in J} \mathcal{R}_{X_j} \right\}$$

Example 21 (extended reals)**.** We defined the space $\overline{\mathbb{R}} \coloneqq [-\infty, \infty]$ of extended reals as the space underlying the measurable space $[-\infty, \infty]$ induced by the distance metric. It is also the coproduct

$$\llbracket A.\ell\, M \rrbracket\, \gamma \coloneqq \iota_\ell \llbracket M\gamma \rrbracket \qquad \llbracket \mathbf{case}\, M\, \mathbf{of}\, \{\ell_\ell x_\ell.M_\ell | \ell \in I\} \rrbracket\, \gamma \coloneqq [\llbracket M_\ell \rrbracket]_{\ell\in I}\, (\gamma, \llbracket M \rrbracket\, \gamma)$$

$$\llbracket \mathbf{<}\ell_1 : M_{\ell_1}, \ldots, \ell_n : M_{\ell_n}\mathbf{>} \rrbracket\, \gamma \coloneqq (\llbracket M_\ell \rrbracket\, \gamma)_{\ell\in\{\ell_1,\ldots,\ell_n\}} \quad \llbracket \mathbf{case}\, M\, \mathbf{of}\, \mathbf{<}\ell : x_\ell | \ell \in I\mathbf{>}.N \rrbracket\, \gamma \coloneqq \llbracket N \rrbracket\, (\gamma, \llbracket M \rrbracket\, \gamma)$$

Fig. 5. Denotations of variant and record terms

$\mathbb{R} \amalg \ulcorner\{\pm\infty\}\urcorner$, i.e., a function $\alpha : \mathbb{R} \to \overline{\mathbb{R}}$ is a random element in the measurable space $\overline{\mathbb{R}}$ iff it is a random element of the coproduct.

Indeed, if α is a random element of the coproduct, then it is a recombination of random elements in $\mathbb{R}$ and $\ulcorner\{\pm\infty\}\urcorner$. A random element β in $\mathbb{R}$ is a measurable function, and since $\mathbb{R}$ is a Borel subset of the measurable space $\overline{\mathbb{R}}$, the function $\lambda r.\beta\, r$ is a classical random element in $\overline{\mathbb{R}}$. A random element in $\ulcorner\{\pm\infty\}\urcorner$ is a σ-simple function, hence also a classical random element in $\overline{\mathbb{R}}$. Therefore α is a recombination of random elements in the measurable space $\overline{\mathbb{R}}$.

Conversely, take a classical random element $\alpha : \mathbb{R} \to \overline{\mathbb{R}}$. These three subsets of the reals are Borel: $E_\pm \coloneqq \alpha^{-1}[\{\pm\infty\}]$ and $E \coloneqq \mathbb{R} \setminus (E_+ \cup E_-)$. If $E = \emptyset$ then $\alpha = \left[E_-.\underline{-\infty}, E_+.\underline{\infty}\right]$. Otherwise, take some $r \in E$. The recombination $\beta \coloneqq [E.\alpha, E^\complement.r] : \mathbb{R} \to \mathbb{R}$ is a random element in $\mathbb{R}$, and $\alpha = \left[E_-.\underline{-\infty}, E_+.\underline{\infty}, E.\beta\right]$. Either way, α is a random element in the coproduct. □

Example 22. The empty type $\mathbf{\{\}}$ denotes the unique space with no elements and no random elements. The unit type denotes a space $\mathbb{1}$ with a single element, the empty tuple $\mathbf{<>}$. This set of elements has a unique metaphorology consisting of the unique constant function. The Booleans denote the discrete space with two points **true**, **false**. Similarly, every enum type $\mathbf{\{}I\mathbf{\}}$, denotes the discrete spaces over its set of labels I. So $\mathbf{Fin}\, n$ denotes the cardinal $\{0, \ldots, n-1\}$. While we have defined immutable arrays and hereditarily finite tensors as functions, their denotations are in a measurable bijective correspondence with products:

$$\llbracket A[n] \rrbracket \cong \llbracket A \rrbracket^n \qquad \llbracket \mathbf{Tensor}\, [n_1, n_2, \ldots, n_k] \rrbracket \cong \mathbb{R}^{n_1 \cdot n_2 \cdots n_k}$$

□

Fig. 5 presents the denotational semantics of terms. It relies on core properties of products and coproducts. For variants, each coproduct of $(X_i)_{i\in I}$ has natural bijections between I-indexed families of measurable functions $f_i : G \times X_i \to Y$ and measurable functions $[f_i]_{i\in I} : G \times \coprod_{i\in I} X_i \to Y$, given by the specification $f_j(\gamma, x) = [f_i]_{i\in I}\, (\gamma, \iota_j x)$ for all $\gamma \in G$, $j \in I$, and $x \in \llcorner X_j \lrcorner$. For records, the product $\prod_{i\in I\uplus J} X_i$ over a disjoint union of indices is their product of products $(\prod_{i\in I} X_i) \times (X_{j\in J} X_j)$.

Example 23. The denotations of the Boolean functions are the usual functions over the Booleans:

$$\begin{aligned} &(\neg) : \mathbb{B} \to \mathbb{B} && (\neg) \coloneqq \lambda\, \{\mathbf{true}.\mathbf{false}; \_.\mathbf{true}\} \\ &(\wedge) : \mathbb{B} \times \mathbb{B} \to \mathbb{B} && (\wedge) \coloneqq \lambda\, \{(\mathbf{true}, \mathbf{true})\, .\mathbf{true}; \_.\mathbf{false}\} \\ &(\Rightarrow) : \mathbb{B} \times \mathbb{B} \to \mathbb{B} && (\Rightarrow) \coloneqq \lambda\, \{(\mathbf{true}, \mathbf{false})\, .\mathbf{false}; \_.\mathbf{true}\} \end{aligned}$$

More generally, the coproduct of every family of discrete spaces is discrete, and every function from the points of any discrete space to the set of points of another space is measurable. A concrete proof for both of these observation is to recall that a space is discrete iff its random elements are the σ-simple functions (cf. Example 12). □

Example 24 (natural numbers as an infinite variant)**.** We will use the coincidence between the natural numbers and semantics of the infinite variant type that has a nullary constructor $\underline{n}$ for every natural number n given by: $\left[\lambda \underline{n}.n\right]_{n\in\mathbb{N}} : \llbracket \{\underline{n} | n \in \mathbb{N}\} \rrbracket \xrightarrow{\cong} \mathbb{N}$. While using this coincidence is mathematically rigorous, it requires us to use reflection extensively. Later, in Sec. 5, we will introduce datatype declarations to our language. These will allow us to internalise inductively defined spaces such as the natural numbers and reduce the reliance on reflection. □

4 FUNCTION SPACES AND SUB-SPACES

We will now use spaces of functions to structure measurability proofs. In the sequel, we will often restrict attention to sub-spaces of functions, and so we will develop the concept of sub-spaces, too. The concepts and results here are not novel, and have appeared in various forms in the Staton et al. [2017] and other existing literature and its supporting material Dash et al. [2023], Ścibior et al. [2018], and Vákár et al. [2019].

4.1 Function spaces

In §3.2 we defined the function space Y^X by equipping the set of measurable functions between two spaces X, Y with the following metaphorology and evaluation function:

$$\llcorner Y^X \lrcorner \coloneqq \mathbf{Qbs}(X, Y) \qquad \mathcal{R}_{Y^X} \coloneqq \{\lambda r.\lambda x.f(r, x) | f : \mathbb{R} \times X \to Y\} \qquad \mathsf{eval} : Y^X \times X \xrightarrow{\lambda(f,x).f\,x} Y$$

Example 25 (random element space)**.** The points of the function space $X^{\mathbb{R}}$ are exactly the random elements in X. Indeed, take any measurable function $f : \mathbb{R} \to X$. Since $\mathrm{id} : \mathbb{R} \to \mathbb{R}$ is classically measurable, it is a random element in $\mathcal{R}_{\mathbb{R}}$. Therefore $f = f \circ \mathrm{id}$ is a random element in X. Conversely, take any random element $\alpha \in \mathcal{R}_X$. Showing $\alpha : \mathbb{R} \to X$ is quasi-measurable amounts to taking a random element $\varphi \in \mathcal{R}_{\mathbb{R}}$ deducing that $\alpha \circ \varphi$ is a random element in X. By definition, $\varphi : \mathbb{R} \to \mathbb{R}$ is Borel measurable, and by the precomposition axiom, $\alpha \circ \varphi$ is a random element.

Specializing this argument to $X \coloneqq \mathbb{R}$, we have the following conservativity result: $f : \mathbb{R} \to \mathbb{R}$ is quasi-measurable iff it is classically measurable (cf. Example 14). Specializing it to $X \coloneqq \mathbb{B}$, we have that $f : \mathbb{R} \to \mathbb{B}$ is quasi-measurable iff it is σ-simple iff the inverse image of **true** is a Borel subset $f^{-1}[\mathbf{true}] \in \mathcal{B}_{\mathbb{R}}$, and equivalently for the inverse image of **false**. More generally, specializing for a discrete space we have that $f : \mathbb{R} \to \ulcorner I \urcorner$ is measurable iff the image f is countable and, and the inverse image of every $i \in I$ is Borel: $f^{-1}[i] \in \mathcal{B}_{\mathbb{R}}$. □

Example 26 (Boolean quantification)**.** Quantification over a countable domain is measurable:

$$\forall, \exists : \mathbb{B}^{\mathbf{Fin}\,n} \to \mathbb{B} \qquad \forall, \exists : \mathbb{B}^{\mathbb{N}} \to \mathbb{B}$$

For the measurability of the infinite operations, take a random element $\alpha = \lambda r.\lambda n.f(r, n)$ for some measurable $f : \mathbb{R} \times \mathbb{N} \to \mathbb{B}$. Then post-composing with universal quantification, for example, is a recombination of random elements:

$$\forall \circ \alpha = (\forall \circ (\lambda r.\lambda n.f(r, n))) = \left[\bigcap_{n \in \mathbb{N}} \{r \in \mathbb{R} | f(r, n) = \mathbf{true}\}\,.\underline{\mathbf{true}}, \bigcup_{n \in \mathbb{N}} \{r \in \mathbb{R} | f(r, n) = \mathbf{false}\}\,.\underline{\mathbf{false}}\right]$$

Each component set in the union has a fixed, constant, natural number as argument, and therefore is a measurable function $\lambda r.f(r, n) : \mathbb{R} \to \mathbb{B}$. By Example 25, it is a random element in $\mathbb{B}$, i.e., a simple function, and so the inverse images of **true** and **false** are Borel. We therefore have a measurable recombination of random elements, and so $\forall \circ \alpha$ is a random element and $\forall$ is measurable.

Existential quantification is also measurable since $\exists : \mathbb{B}^{\mathbb{N}} \xrightarrow{\lambda f.\neg \circ f} \mathbb{B}^{\mathbb{N}} \xrightarrow{\forall} \mathbb{B} \xrightarrow{\neg} \mathbb{B}$. Finite quantification is measurable by a similar argument, or by reducing the proof to infinite quantification. □

Example 27 (countable minimisation and selection)**.** The functions $\min, \operatorname{argmin} : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ are measurable. Since $\min f = f(\operatorname{argmin} f)$, we need to prove the selection operator argmin measurable. Indeed, take a random element $\alpha \in \mathcal{R}_{\mathbb{N}^{\mathbb{N}}}$, noting this set is Borel:

$$(\operatorname{argmin} \circ \alpha)^{-1}[k] = \{r \in \mathbb{R} | \alpha\, r\, n = k \land \forall i < n.\alpha\, r\, n \neq k\} \in \mathcal{B}_{\mathbb{R}} \qquad \square$$

Example 28 (sequential extrema)**.** The extremum operations $\sup, \inf : \overline{\mathbb{R}}^{\mathbb{Q}} \to \overline{\mathbb{R}}$ are measurable. Since $\inf \vec{x} = -\sup{(-x_n)}_n$, it is measurable by type-checking. For the supremum, take a random element $\alpha \in \mathcal{R}_{\overline{\mathbb{R}}}$, and so $\alpha = \lambda r.\,(\alpha_n)_n$ for a sequence of random elements $\vec{\alpha} \in \mathcal{R}_{\overline{\mathbb{R}}}^{\mathbb{N}}$. We need to show that $\sup \circ \alpha : \mathbb{R} \to \overline{\mathbb{R}}$ is a random element of $\mathcal{R}_{\overline{\mathbb{R}}}$, i.e., a classically measurable function. The Borel sets of $\overline{\mathbb{R}}$ are generated by the intervals $[-\infty, a)$ for $a \in \overline{\mathbb{R}}$, and we have $\sup^{-1}[-\infty, a) = \{\vec{x} | \forall n. x_n \leq a\}$ and the predicate in this subset is Boolean by type-checking. □

Example 29 (limit superior and inferior)**.** The operations $\limsup, \liminf : \overline{\mathbb{R}}^{\mathbb{N}} \to \overline{\mathbb{R}}$ are measurable. We cannot directly deduce this fact from Example 28, since the extremum ranges over a potentially uncountable set: $\limsup \vec{x} \coloneqq \sup \left\{\lim_k x_{n_k} \middle| {(x_{n_k})}_k \text{ is a converging subsequence}\right\}$. However, we can prove measurability directly. Take a random element $\alpha \in \mathcal{R}_{\overline{\mathbb{R}}^{\mathbb{N}}}$, and $a \in \overline{\mathbb{R}}$:

$$(\limsup \circ \alpha)^{-1}[-\infty, a] = \{r | \forall n. \exists k. k > n \wedge x_k \leq a\}$$

The function that takes pointwise limit superior of measurable function sequences is measurable:

$$\limsup : (\mathbb{R}^{\mathbb{R}})^{\mathbb{N}} \to \mathbb{R}^{\mathbb{R}} \qquad \limsup_n f_n = \lambda x. \limsup_n f_n\, x$$ □

4.2 Sub-spaces

Let X be a space and $S \subseteq \llcorner X \lrcorner$ be a set of points. We can equip S with the largest metaphorology that makes the inclusion $\lambda x.x : S \subseteq X$ into a measurable function:

$$\mathcal{R}_S \coloneqq \{\alpha : \mathbb{R} \to S | \lambda r.\alpha(r) \in \mathcal{R}_X\}$$

Generalising away from the specifics of set-inclusion, given a measurable function $e : X \to Y$, we say e is a *subspace embedding*[2], and write $e : X \hookrightarrow Y$, when:

- $e : \llcorner X \lrcorner \to \llcorner Y \lrcorner$ is injective, i.e., for $x, y \in \llcorner X \lrcorner$: $ex = ey \implies x = y$; and
- post-composing e reflects random elements, i.e., for $\alpha : \mathbb{R} \to \llcorner X \lrcorner$: $e \circ \alpha \in \mathcal{R}_Y \implies \alpha \in \mathcal{R}_X$.

Example 30. Every subspace inclusion $\lambda x.x : S \to X$ is a subspace embedding. □

Example 31 (subspaces of Euclidean spaces)**.** Subsets of Euclidean spaces, such as $[a, b) \subseteq \mathbb{R}$, induce subspaces in two different ways: either as the space underlying a measurable subspace $[a, b)$ of the measurable space $\mathbb{R}$, or as the subspace $[a, b)$ of the space underlying the measurable space $\mathbb{R}$. These two notions of subspace coincide, and so the many examples we encountered so far for subspaces of measurable spaces are all examples for subspace embeddings. □

Example 32 (subspace classifier)**.** Let **Prop** be the indiscrete space $[\{\mathbf{true}, \mathbf{false}\}]$ over the Booleans. The constantly **true** function $\underline{\mathbf{true}} : \mathbb{1} \to \mathbf{Prop}$ is a subspace embedding. Given a space X, every function $\varphi : \llcorner X \lrcorner \to \{\mathbf{true}, \mathbf{false}\}$ is then a measurable function $\varphi : X \to \mathbf{Prop}$, and it induces a subspace embedding $\{x \in \llcorner X \lrcorner | \varphi x = \mathbf{true}\} \hookrightarrow X$. □

Example 33 (function subspaces)**.** We will study spaces of functions with additional properties:

- For any interval $I \subseteq \mathbb{R}$, let $\mathrm{C}_0 I \coloneqq \{f : I \to [-\infty, \infty] | f \text{ continuous}\} \hookrightarrow [-\infty, \infty]^I$ be the space of continuous functions. We can generalise from I to arbitrary topological spaces A as a subspace of functions from the (quasi-Borel) space underlying A.
- $\{f : I \to [-\infty, \infty] | f \text{ differentiable in } I\} \hookrightarrow \mathrm{C}_0 I$.
- $\mathrm{C}_1 I \coloneqq \{f : I \to [-\infty, \infty] | f \text{ continuously differentiable in } I\} \hookrightarrow \mathrm{C}_0 I$.

[2]⚠ Categorically, the subspace embeddings are the strong monomorphisms.

and so on for any other property of interest. All of these spaces $F \hookrightarrow \mathbb{R}^X$ have a measurable evaluation function $\mathsf{eval} : F \times X \xrightarrow{\lambda(f,x).(f,x)} \mathbb{R}^I \times X \xrightarrow{\lambda(f,x).f\,x} [-\infty, \infty]$. □

Example 34 (spaces of sequences)**.** We'll call subspaces of the space of sequences $[-\infty, \infty]^{\mathbb{N}}$ *sequential* spaces. We will use these sequential spaces:

- Converging sequences: $\mathbf{Converge} \coloneqq \{\vec{x} | \exists \lim_{n\to\infty} x_n\}$;
- Cauchy sequences: $\mathbf{Cauchy} \coloneqq \mathbf{Converge} \cap \mathbb{R}^{\mathbb{N}}$;
- Vanishing sequences: $\mathbf{Vanish} \coloneqq \{\vec{x} \in \mathbf{Converge} | \lim_{n\to\infty} x_n = 0\}$;
- Convergence rates: $\mathbf{Rate} \coloneqq \mathbf{Vanish} \cap (0, \infty]^{\mathbb{N}}$; and
- Sequences converging at rate $\vec{\triangle} \in \mathbf{Rate}$: $\{\vec{x} \in \mathbf{Converge} | \forall n. |x_n - \lim_k x_k| < \triangle_n\}$

Since $\lim \vec{x} = \limsup \vec{x}$ for every $\vec{x} \in \mathbf{Converge}$, taking limits of sequences are measurable operations $\lim : \mathbf{Converge} \to \overline{\mathbb{R}}$ and $\lim : \mathbf{Cauchy} \to \mathbb{R}$. □

In order to define a measurable function $f : X \to Y$ whose codomain is a subspace $e : Y \hookrightarrow Z$, we can first define a measurable function $g : Y \to Z$, and then prove that the function g lifts along the embedding: for every $x \in \llcorner X \lrcorner$, there exists a (necessarily unique) $y \in \llcorner Y \lrcorner$ with $g\,x = e\,y$.

Example 35. The rational approximation function $\mathsf{approx} : \mathbb{R} \to \mathbb{Q}^{\mathbb{N}}$ from the introduction lifts along the subspace embedding from $\mathbf{Converge} \cap \mathbb{Q}^{\mathbb{N}}$. It restricts further along the embedding from the space of sequence that converge at $\left(\frac{1}{n}\right)_n \in \mathbf{Rate}$. □

Example 36 (limits and derivatives)**.** Let $L \hookrightarrow \mathbb{R}^{\mathbb{R}} \times \mathbb{R}$ be the subspace of pairs (f, a) of measurable functions $f : \mathbb{R} \to \mathbb{R}$ that have a limit at the given point $a \in \mathbb{R}$. Then taking the limit of the function at this point is a measurable operation:

$$\lambda\,(f, a)\,.\,\lim_{x\to a} f\,x \qquad \lim_{x\to a} f\,x = \lim_{n\to\infty} f(\mathsf{approx}\;a)_n$$

Similarly, let $D \hookrightarrow \mathbb{R}^{\mathbb{R}} \times \mathbb{R}$ be the space of functions with a point at which this function has a derivative. Then taking the derivative of this function is measurable:

$$\frac{\mathrm{d}}{\mathrm{d}}\Big| : D \to \mathbb{R} \qquad \frac{f\,x}{\mathrm{d}x}\Big|_a = \lim_{h\to 0} \frac{f(a+h) - f(a)}{h}$$

In particular, let $\mathbf{C}_1 \hookrightarrow \mathbb{R}^{\mathbb{R}}$ be the subspace of differentiable functions with a smooth derivative. Then differentiation is a measurable function $\frac{\mathrm{d}}{\mathrm{d}}\Big| : \mathbf{C}_1 \to \mathbf{C}_0$. □

An embedding $e : X \hookrightarrow Y$ determines a measurable isomorphism between the embedded space X and its image $e[X]$. To conclude this section, lets look at a non-embedding:

Non-example 37. The injective inclusion function $\lambda x.x : \mathbb{B} \to \mathbf{Prop}$ is measurable, but not a subspace embedding. Were it a subspace embedding, it would be a measurable isomorphism and in particular the two metaphorologies would be in bijection. The metaphorology $\mathcal{R}_{\mathbb{B}}$ consists of the σ-simple functions, which are in bijection with Borel subsets of reals, of which we have a continuum. Whereas $\mathcal{R}_{\mathbf{Prop}}$ consists of all set-theoretic functions $\mathbb{R} \to \{\mathbf{true}, \mathbf{false}\}$, and by Cantor's theorem it cannot be in bijection with $\mathcal{R}_{\mathbb{B}}$. □

The distinction between a **Prop**-valued predicates and a Boolean predicate is subtle. It is similar to the distinction between a predicate and a decidable/computable/constructive predicate, although in our highly non-constructive setting it means, broadly, measurable. We will use this distinction to develop a simply-typed account of measurable events.

4.3 Event subspaces

It is time to derive the notion of measurable event. An *event* in a space X is a subset $E \subseteq \llcorner X \lrcorner$ whose inverse images under random elements are Borel: for every $\alpha \in \mathcal{R}_X$, we have $\alpha^{-1}[E] \in \mathcal{B}_{\mathbb{R}}$. A subset E is an event iff its characteristic function $[- \in E] : E \to \mathbf{Prop}$ lifts along the inclusion $\mathbb{B} \rightarrowtail \mathbf{Prop}$. We denote the set of events by $\mathcal{B}_X$.

Example 38. The events in the space $\mathbb{R}$ are exactly the classical Borel subsets. To see why, take a classical Borel subset E. Each random element in $\mathbb{R}$ is a Borel measurable function $\alpha : \mathbb{R} \to \mathbb{R}$, and so $\alpha^{-1}[E]$ is indeed a Borel subset, and so E is an event. Conversely, if E is an event, since $\mathrm{id} : \mathbb{R} \to \mathbb{R}$ is a random element in $\mathcal{R}_{\mathbb{R}}$, we deduce that E must be a Borel subset. □

Example 39. The set of convergence rates is an event in the space of vanishing sequence, since its characteristic function is measurable into $\mathbb{B}$: $[\vec{\triangle} \in \mathbf{Vanish}] = \forall n.\triangle_n > 0$. The set of Cauchy sequences is an event in the space of all sequences:

$$[\vec{x} \in \mathbf{Cauchy}] = \forall \varepsilon \in \mathbb{Q}.\varepsilon > 0 \Rightarrow \exists n \in \mathbb{N}.\forall m, k \in \mathbb{N}.\, |x_{n+m} - x_{n+k}| < \varepsilon$$

Here we needed to adapt the usual definition for a Cauchy sequence from ranging over all positive reals to ranging over positive rationals. Since the adapted predicate still characterises Cauchy sequences, the set of Cauchy sequences is indeed an event. We have suppressed the chosen enumeration of rationals that allows us to quantify over rationals. □

A *Borel embedding* $e : X \hookrightarrow Y$ is a subspace embedding $e : X \hookrightarrow Y$ onto an event $e[X] \in \mathcal{B}_Y$.

Example 40 (concrete embeddings)**.** The real line $\mathbb{R}$ embeds as an event in the extended real line $[-\infty, \infty]$. All of the examples for concrete subspaces, such as $[a, b)$, $\mathbb{Q}$, etc., have Borel embeddings into the real line via inclusion. Since equality testing and comparison of real numbers are measurable: $(=), (\leq), (<) : [-\infty, \infty]^2 \to \mathbb{B}$, events include important subsets, e.g., the: unit circle $\left\{(x, y) \in \mathbb{R}^2 \middle| x^2 + y^2 = 1\right\}$; diagonal $\{(x, y) \in \mathbb{R} | x = y\}$; half-planes $\{(x, y) \in \mathbb{R} | x < y\}$; etc. □

Example 41. The sequential spaces from Example 34 are a chain of Borel embeddings:

$$\triangle\text{-}\mathbf{Converge} \hookrightarrow \mathbf{Cauchy} \hookrightarrow \mathbf{Converge} \hookrightarrow \overline{\mathbb{R}}^{\mathbb{N}}$$

For the left-most embedding, the actual convergence rate of a sequence is a measurable operation:

$$\mathbf{rate} : \mathbf{Cauchy} \to [0, \infty) \qquad \mathbf{rate}\, \vec{x} \coloneqq \left(\sup_{k \in \mathbb{N}} |x_{n+k} - \lim \vec{x}|\right)_{n \in \mathbb{N}}$$

and checking whether this rate is strictly below a given rate is measurable $\forall n.(\mathbf{rate}\, \vec{x})_n < \triangle_n$.

For the middle embedding, note that a converging sequence $\vec{x}$ is Cauchy iff $\lim \vec{x} \neq \pm\infty$. For the right-most embedding, note that a sequence converges iff $\limsup \vec{x} = \liminf \vec{x}$. □

Example 42. Every coproduct injection is a Borel embedding $\iota_j : X_j \hookrightarrow \coprod_{i \in I} X_i$. □

Example 43 (recombination)**.** Countable measurable recombinations of measurable functions are measurable. Explicity, consider a countable partition $\llcorner X \lrcorner = \biguplus_{i \in I} E_i$ of the points of a space X into a sequence of events $(E_i)_i \in \mathcal{B}_X^I$ and an I-indexed family of measurable functions $f_i : E_i \to Y$. Then the recombination function $f \coloneqq [E_i.f_i]_{i \in I} : X \to Y$ is measurable. Indeed, given a random element $\alpha \in \mathcal{R}_X$, we have that $F_i \coloneqq \alpha^{-1}[E_i] \in \mathcal{B}$ and partitions $\mathbb{R}$. Since each $f_i \circ \alpha$ is a random element of Y, their recombination $[F_i.f_i \circ \alpha]_{i \in I} = f \circ \alpha$ is a random element. □

Example 44 (measurable boxes)**.** Let $(E_i)_{i \in I}$ be a countable family of events in a countable family of spaces $(X_i)_i$. Then their Cartesian product is an event in the product space $\prod_{i \in I} E_i \in \mathcal{B}_{\prod_i X_i}$. Indeed, since a random element α of $\prod_i X_i$ is a random element component-wise, we have that the inverse image is $\alpha^{-1}[\prod_i E_i] = \cap_{i \in I} \alpha^{-1}[E_i] \in \mathcal{B}_{\mathbb{R}}$. The product may, however, contain events that are not generated by the measurable boxes, even for binary products (see Example 49). □

Example 45 (event classifier and the space of Borel subsets)**.** The constantly true function is a Borel embedding $\underline{\mathbf{true}} : \mathbb{1} \to \mathbb{B}$. In analogy with the subspace classifier, this embedding classifies Borel embeddings. As a consequence, we can equip the set of events of a space X with the metaphorology of the function space $\mathbb{B}^X$, transported along the bijection between events and measurable functions given via characteristic functions. We can characterise it explicitly as:

$$(\mathcal{U} : \mathbb{R} \to \mathcal{B}_X) \in \mathcal{R}_{\mathcal{B}_X} \iff \forall \alpha \in \mathcal{R}_X . \{(r, s) \in \mathbb{R} \times \mathbb{R} | \alpha\, s \in \mathcal{U}\, r\} \in \mathcal{B}_{\mathbb{R}\times\mathbb{R}}$$

As one may expect, measurability-by-type-checking proofs are easier to validate than working directly with this definition. Note in passing that we can continue to form spaces of Borel subsets iteratively. For example, Sabok et al.'s [2021] use the space $\mathcal{B}_{\mathcal{B}_{\mathbb{R}}}$. Its points are known as the Borel-on-Borel subsets in classical descriptive set theory. Sabok et al. use measurability conditions to rule out undesirable observational behaviour in the semantics of name generation and obtain more accurate programming language models. □

Example 46 (free measurable spaces over a space)**.** The events $\mathcal{B}_X$ in a space X form a σ-field. The *free* measurable space $\ulcorner X \urcorner^{\mathbf{Meas}}$ over X has the same points and the events $\mathcal{B}_X$ as its σ-field. By Example 38, the free measurable space over the reals is the standard measurable space structure, the σ-field of Borel subsets.

Given a measurable space A and a space X, we have the relationship:

$$f : X \to \llcorner A \lrcorner_{\mathbf{Qbs}} \text{ quasi-measurable} \iff f : \ulcorner X \urcorner^{\mathbf{Meas}} \to A \text{ classically measurable}$$

From this relationship we can derive that the free measurable space construct preserves coproducts of spaces, and that the space underlying the product of two measurable spaces is the product of their underlying spaces. We mentioned this preservation of products in §3.2 to derive the quasi-measurability of multi-argument functions between measurable spaces. □

Example 47. The σ-field operations are measurable functions over the space of events, as the pointwise Boolean operations classify them. For example:

$$\begin{array}{lll} (-^{\complement}) : \mathcal{B}_X \to \mathcal{B}_X & (\cap) : \mathcal{B}_X^2 \to \mathcal{B}_X & (\bigcup) : \mathcal{B}_X^{\mathbb{N}} \to \mathcal{B}_X \\ [x \in E^{\complement}] := \neg[x \in E] & [x \in E \cap F] := [x \in E] \wedge [x \in F] & [x \in \bigcup_{n\in\mathbb{N}} E_n] := \exists n.[x \in E_n] \end{array}$$

With these operations we derive other standard events, for example outcomes that belong eventually in all positions in a series of events (limit inferior, safety properties), and outcomes that belong infinitely often (limit superior, liveness properties):

$$\limsup, \liminf : \mathcal{B}_X^{\mathbb{N}} \to \mathcal{B}_X \qquad \limsup_{n\to\infty} E_n := \bigcup_k \bigcap_{\ell=k}^{\infty} E_\ell \qquad \limsup_{n\to\infty} E_n := \bigcap_k \bigcup_{\ell=k}^{\infty} E_\ell$$

Every measurable function $f : X \to Y$ induces a measurable operation on events via inverse image $f^{-1} : \mathcal{B}_Y \to \mathcal{B}_X$. This operation is measurable not only in the event we send to its inverse image, but also in the function, since its classifying predicate is measurable by type-checking:

$$\lambda (f, E) . f^{-1}[E] : Y^X \times \mathcal{B}_Y \to \mathcal{B}_X \qquad [x \in f^{-1}[E]] := [f\, x \in E]$$

For example, the set of outcomes in an infinite sequence in which a given event occurs constantly in some suffix or infinitely often are events:

$$\text{-eventually}, \text{-i.o.} : \mathcal{B}_X \to \mathcal{B}_{X^{\mathbb{N}}} \quad E\text{-eventually} := \liminf_{n\to\infty} {\pi_n}^{-1}[E] \quad E\text{-i.o.} := \limsup_{n\to\infty} {\pi_n}^{-1}[E] \qquad □$$

Non-example 48. Not all relevant operations on Borel subsets are measurable. Sabok et al. [2021] show that emptiness checking is not a measurable function $(= \emptyset) : \mathcal{B}_{\mathbb{R}} \to \mathbb{B}$. By defining appropriate reductions, other relevant operations and properties are not measurable either, including: subset inclusion $(\subseteq) : \mathcal{B}_{\mathbb{R}}^2 \to \mathbb{B}$; testing for topological properties such as being open or closed subsets; having finite or countable cardinality; and so on. □

Example 49 (Aumann's theorem)**.** A frequently asked question is whether the free measurable space construction allows us to circumvent Aumann's result (Theorem 2). The situation is subtle. We can form the free measurable space over $\mathbb{R}^{\mathbb{R}}$ and we do have a classically measurable counterpart to the evaluation function $\mathsf{eval} : \ulcorner \mathbb{R}^{\mathbb{R}} \times \mathbb{R} \urcorner \to \ulcorner \mathbb{R} \urcorner = \mathbb{R}$. We also have a measurable function $\lambda (f, x) . (f, x) : \ulcorner \mathbb{R}^{\mathbb{R}} \times \mathbb{R} \urcorner \to \ulcorner \mathbb{R}^{\mathbb{R}} \urcorner \times \mathbb{R}$, but it is *not* a measurable isomorphism. Summarising:

$$\begin{tikzcd}
\ulcorner \mathbb{R}^{\mathbb{R}} \times \mathbb{R} \urcorner \arrow[rr, "\mathsf{eval}"] \arrow[dr, "\lambda (f, x) . (f, x)"'] & {} \arrow[d, phantom, "="] & \mathbb{R} \\
& \ulcorner \mathbb{R}^{\mathbb{R}} \urcorner \times \mathbb{R} \arrow[ur, "\nexists \text{(by Aumann's theorem)}"'] &
\end{tikzcd}$$

The σ-field of the measurable space $\ulcorner \mathbb{R}^{\mathbb{R}} \times \mathbb{R} \urcorner$ has strictly more measurable subsets than the product measurable space $\ulcorner \mathbb{R}^{\mathbb{R}} \urcorner \times \mathbb{R}$. These additional subsets make evaluation measurable. □

4.4 Partiality

By Example 42, every space X embeds $\downarrow : X \hookrightarrow X_\perp$ into an accompanying space of *partial* outcomes, which adjoins a new point $\perp$, representing the undefined outcome:

$$X_\perp \coloneqq [\![\{\perp, \downarrow : X\}]\!] \cong \{\perp\} \amalg X$$

The space $(Y)_\perp^X$ then represents the space of partial functions $X \rightharpoonup Y$. To define a partial function, we exhibit its domain of definition as an event, sending every point outside this domain to $\perp$:

Example 50. Division $\lambda (x, y) . \frac{x}{y} : \mathbb{R} \times \mathbb{R} \rightharpoonup \mathbb{R}$ is a partial function, its domain of definition is the Borel subset $\mathbb{R} \times (\mathbb{R} \setminus \{0\})$. Formally, we can define:

$$\tfrac{x}{y} \coloneqq \mathbf{case}\ y = 0\ \mathbf{of}\ \{\mathbf{true}. \downarrow \tfrac{x}{y}; \mathbf{false}.\perp\}$$ □

More formally, the domain of definition of a partial function $f : X \rightharpoonup Y$ is the subspace induced by the inverse image of Y: $\mathrm{Dom} f \coloneqq f^{-1} [\{\downarrow y | y \in Y\}]$. It is an event by type-checking, and so the inclusion yields a Borel embedding $\mathrm{Dom} f \hookrightarrow X$. When the domain of definition is the entire space X, we say that f is *total*. Post-composition with $\downarrow : Y \to Y_\perp$ then gives a subspace embedding of the ordinary function space into the partial-function space: $(\downarrow \circ) : Y^X \hookrightarrow Y_\perp^X$. It may not be Borel.

Example 51. Taking the limit of a sequence is a partial operation $\lim : \overline{\mathbb{R}}^{\mathbb{N}} \rightharpoonup \overline{\mathbb{R}}$. Its domain of definition is the Borel subset of converging sequences $\mathrm{Dom}\ \lim = \mathbf{Converge}$. □

We can compose partial functions, and the measurability of countable intersections and inverse images (Example 47) ensure the corresponding domains of definition are events.

Example 52. We can make this treatment of partial functions fully formal by defining partial function composition $(\circ) : Z_\perp^Y \times Y_\perp^X \to Z_\perp^X$, or even by defining a suitable type-theory and its semantics. We will not do so here. □

Example 53 (Borel embeddings as partial-map sections)**.** Let $e : X \hookrightarrow Y$ be a Borel embedding. This embedding has a canonical one-sided inverse partial function $p : Y \rightharpoonup X$. Indeed, since e is an embedding, the function $e' \coloneqq (\lambda x.e\, x) : X \to E$ is a measurable isomorphism. We then define

$p : Y \rightharpoonup X$ with $\mathrm{Dom} p = E$ and $p\,y \coloneqq (e')^{-1}y$. We then have that the composition of the two partial functions $p \circ e$ is the total function $\mathrm{id} : X \to X$. In the converse direction, the composite $e \circ p : Y \rightharpoonup Y$ is not the identity, but its domain of definition is E and it lifts to the identity on E along the embedding $E \subseteq Y$, i.e.: $(e \circ p)y = y$ for every $y \in E$.

Conversely, given a pair of partial functions $X \xrightharpoonup{e} Y \xrightharpoonup{p} X$ such that: $p \circ e$ is total and the identity on X; and $e \circ p$ restricts to the identity on its domain of definition, then e is total and is a Borel embedding $X \hookrightarrow Y$, and p is the one-sided inverse we defined in the previous paragraph. □

Example 54. The composition of Borel embeddings $e_1 : X \hookrightarrow Y$ and $e_2 : Y \hookrightarrow Z$ is a Borel embedding $e \coloneqq e_2 \circ e1 : X \to Z$. Indeed, by Example 53, consider the corresponding right-sided inverses $p_1 : Y \rightharpoonup X$ and $p_2 : Z \rightharpoonup Y$, and then note:

$$p_2 \circ p_1 \circ e_2 \circ e_1 = p_2 \circ e_1 = \mathrm{id} \qquad \text{and} \qquad \mathrm{Dom}(e_2 \circ e_1 \circ p_1 \circ p_2) = \mathrm{Dom} p_1 \cap {p_1}^{-1}[\mathrm{Dom} p_2]$$

and so $e_2 \circ e_1 \circ p_1 \circ p_2$ does restrict to the identity on its domain of definition. □

Example 55 (pullbacks)**.** Given a *cospan* of measurable functions $X \xrightarrow{f} Z \xleftarrow{g} Y$, then their *pullback* is the following sub-space equipped with the restriction of the two projections:

$$f \bowtie g \coloneqq \{(x, y) \in X \times Y | fx = gy\} \qquad X \xleftarrow{\pi_1} f \bowtie g \xrightarrow{\pi_2} Y$$

When f is an embedding, so is π_1, and when f is a Borel embedding, so is π_1, and symmetrically for g and π_2. Indeed, if f is an embedding, then π_2 is injective. Take a putative random element $\gamma = \lambda r.(\alpha r, \beta r)$ of $f \bowtie g$, such that β is a random element of Y. Then $f \circ \alpha = g \circ \beta$ is a random element in Z, and since f is an embedding, α is a random element in X, and so γ is a random element in the pullback. If, moreover, f is a Borel embedding, then we have a Boolean predicate $\lambda z.(z \in f[X]) : Z \to \mathbb{B}$ classifying $f[X]$. Then $\lambda y.(gy \in f[X]) : Y \to \mathbb{B}$ classifies $\pi_2[f \bowtie g]$. □

5 SELECTION PRINCIPLES AND DATATYPES

We will sometimes use intermediate data-structures in measurability proofs. Some statistical models can be expressed more naturally with when the outcome space is denoted by a hierarchical data type. Some embedding theorems, e.g., use coinductive datatypes for Cantor and Lusin schemes. We introduce datatypes by constructing selection principles: measurable functions that select an element satisfying some specification when such an element exists. The inductive fragment is already present in Ścibior et al. [2018]. Our use of induction and coinduction separately allows us to avoid the domain theory of Vákár et al. [2019], at the price of a more complicated type system.

5.1 Selection principles

We already encountered a couple of selection principles. In Example 27 we showed there is a measurable function $\mathrm{argmin} : \mathbb{N}^{\mathbb{N}} \to \mathbb{N}$ that selects an index i for which the given function attains its minimum. In the introduction we showed a measurable function that selects an approximating rational sequence for every real number. The selection can be unique given additional specification, for example the argmin function chooses the smallest index. The selection can also be parameterised by some additional parameter, such as a given enumeration of the rationals.

A measurable function $\wr : X \to Y$ is a *selection principle* for a property $\varphi : X \times Y \to \mathbf{Prop}$ when $\forall x \in X.\varphi(x, \wr x)$. It follows then that $\forall x \in X.\exists y \in Y \varphi(x, y)$, but moreover we can choose a witness y measurably. We also say that $\wr$ *selects* for φ.

Example 56. The function $\mathrm{argmin} : \mathbb{N}^{\mathbb{N}}$ selects for the predicate $\lambda\,(f, i)\,.\forall \mathrm{j}.fi \le fj$. The function $\min : \mathbb{N}^{\mathbb{N}}$ selects for the prediate $\lambda\,(f, n)\,.n \le fj$. The function argmin also selects for the stronger predicate that states the selected index is minimal:

$$\lambda\,(f,i)\,.(\forall j.fi \le fj) \wedge \forall i'.(\forall j.fi' \le fj) \implies i \le i' \qquad \square$$

Example 57. The function $\mathsf{approx} : \mathbb{R} \to \mathbb{Q}^{\mathbb{N}}$ from the introduction selects for the properties: $\lambda\,(r,\vec{q})\,.\lim \vec{q} = r$; the convergence rate is $\left(\frac{1}{n}\right)_n$; the convergence is from below; etc. $\square$

In Example 10 we defined approx using an enumeration $q : \mathbb{N} \twoheadrightarrow \mathbb{Q}$ for the rationals. With datatypes we can similarly require as parameter similar scaffolding or, better, define it measurably.

Example 58. We define a general selection principle that selects a position in a sequence satisfying a given event, where both the sequence and the event can vary measurably:

$$\text{Ƨ} : \left\{(\vec{x},E) \in X^{\mathbb{N}} \times \mathcal{B}_X \middle| \exists n.x_n \in E\right\} \to \mathbb{N} \qquad \text{selecting for} \qquad \lambda\,((\vec{x},E),n)\,.x_n \in E$$

The domain subspace, given by a measurable predicate, is a Borel subspace $D \hookrightarrow X^{\mathbb{N}} \times \mathcal{B}_X$. Define a partial function $\text{Ƨ} : X^{\mathbb{N}} \times \mathcal{B}_X \rightharpoonup \mathbb{N}$, with $\mathrm{Dom}(\text{Ƨ}) = D$, by $\text{Ƨ}\,(\vec{x},E) = \min\{n | x_n \in E\}$. For measurability, take a random element $\beta = (\lambda r.\left((\alpha_n r)_{n\in\mathbb{N}}, E\,r\right)) \in \mathcal{R}_{X^{\mathbb{N}}\times\mathcal{B}_X}$, $\beta[\mathbb{R}] \subseteq D$, and $n \in \mathbb{N}$:

$$(\text{Ƨ} \circ \beta)^{-1}[\{n\}] = \{r | \alpha_n r \in E\,r, \forall i.i < n \implies \alpha_n r \notin E\}$$

We can use Ƨ either as a partial function or as total function over its domain of definition. Moreover:

$$(\lambda\,(\vec{x},E)\,.\left(x_{\text{Ƨ}(\vec{x},E)}, E\right)) : \left\{(\vec{x},E) \in X^{\mathbb{N}} \times \mathcal{B}_X \middle| \exists n.x_n \in E\right\} \to \{(x,E) | x \in E\} \qquad \square$$

Notation for function specification. As we define selection principles for more complicated events, the notation we used in Example 58 to declare the type of Ƨ gets heavy. Instead, we adopt the following a Hoare-triple notation that a function relies on a certain pre-condition and guarantees a certain post-condition. We write, glossing over the difference between typing contexts and records:

$$\frac{\Gamma \vdash \varphi : \mathbf{Prop} \qquad \Gamma,\Delta \vdash \psi : \mathbf{Prop}}{f : \{\Gamma \vdash \varphi\} \to \{\Delta \vdash \psi\}} \quad \text{meaning: } f : \{\Gamma \vdash \varphi\} \to \Delta \text{ such that } \forall \gamma \in \Gamma.\varphi(\gamma) \Rightarrow \psi(\gamma, f\,\gamma)$$

Going back to Example 58, we write:

$$\text{Ƨ} : \left\{(\vec{x},E) \in X^{\mathbb{N}} \times \mathcal{B}_X \middle| \exists n.x_n \in E\right\} \to \{n : \mathbb{N} | x_n \in E\}$$

We sometimes need the stronger specification—we select the earliest position in the sequence:

$$\text{Ƨ} : \left\{(\vec{x},E) \in X^{\mathbb{N}} \times \mathcal{B}_X \middle| \exists n.x_n \in E\right\} \to \{n : \mathbb{N} | x_n \in E, \forall i < n.x_i \notin E\}$$

5.2 Inductive datatypes by example

Our type theory includes *type families/schema*. For example, we will later define the *list* family $\alpha \vdash \mathbf{List}\,\alpha : \mathbf{Type}$. We can instantiate this family to get concrete types such as lists of Booleans ($\mathbf{List}\,\mathbb{B}$) and lists of naturals ($\mathbf{List}\,\mathbb{N}$). We may exhibit some existing spaces, such as the natural numbers space or sequential spaces ($A^{\mathbb{N}}$, as inductive or coinductive datatypes, allowing us to use induction and coinduction to define measurable functions into and out of these spaces.

Example 59 (lists)**.** Lists are an inductive datatype, consisting of the empty list $\mathbf{Nil} : \mathbf{List}\,A$ and an inductive data constructor that appends an element $x : A$ at the *head* of a list *tail* $\vec{x} : A$, resulting in the list $(x :- \vec{x}) : \mathbf{List}\,A$. We desugar $[x_1, x_2 \ldots, x_n]$ to the list $x_1 :- x_2 :- \ldots :- x_n :- \mathbf{Nil}$. Induction over lists requires a function that defines the result in these two cases: when the list is empty; and given the head of the list and the inductively defined result for its tail. For example:

$$\mathbf{length} : \mathbf{List}\,A \to \mathbb{N} \qquad \mathbf{length}\,\vec{x} = \mathbf{List}\,A.\mathbf{fold}\,\lambda\,\{\mathbf{Nil}.0; x :- n.n+1\}\vec{x} \qquad \text{(list length)}$$

$$(+\!\!+) : (\mathbf{List}\,A)^2 \to \mathbf{List}\,A \qquad \vec{x} +\!\!+ \vec{y} = \mathbf{List}\,A.\mathbf{fold}\,\lambda\,\{\mathbf{Nil}.\vec{y}; x :- \vec{z}.x :- \vec{z}\}\vec{x} \qquad \text{(concatenation)}$$

The natural numbers are also denoted by an inductive type. Although often we can use their discreteness and deduce that a given function is measurable, we can also define measurable functions by induction. For example, the function $\lambda n.[1, \ldots, n]$ that enumerates the numbers from 1 to n:

$$[1, \ldots, -] : \mathbb{N} \to \mathbf{List}\,\mathbb{N} \qquad [1, \ldots, n] = \mathbb{N}.\mathbf{fold}\,\lambda\left\{0.\underline{\mathbf{Nil}}; 1 + f.\lambda m.(1+m) :- f(1+m)\right\} n\, 0$$

Lists, like every type family, has an accompanying *map* operation, that applies the given functions in covariant or contravariant positions: $\mathbf{List}.\mathbf{map}\ (\lambda x.x^2)[1,2,3] = [1,4,9]$. E.g., tabulation:

$$[-i|1 \le i \le -]\ : X^{\mathbb{N}} \times \mathbb{N} \to \mathbf{List}\,X \qquad [f i|1 \le i \le n] \coloneqq \mathbf{List}\,X.\mathbf{map}\,f\,[1, \ldots, n]$$

The 'snoc' operation $(-:) : \mathbf{List}\,A \times A \to \mathbf{List}\,A$, appends on the right: $\vec{x} -: y \coloneqq \vec{x} +\!\!+ [y]$. □

Example 60 (selecting from a list)**.** We can select from lists by iterating through their elements:

$$\text{Ƨ} : \{(\vec{x}, E) \in \mathbf{List}\,X \times \mathcal{B}_X | \exists i.1 \le i \le \mathbf{length}\,\vec{x} \wedge x_i \in E\} \to \{x : X | x \in E\}$$

We do so by defining it as a partial function:

$$\text{Ƨ}\,(\vec{x}, E) \coloneqq \mathbf{List}\,.\mathbf{fold}\,\lambda\,\{\mathbf{Nil}.\bot; y :- z.\mathbf{case}\,y \in E\,\mathbf{of}\,\{\mathbf{true}.\downarrow y; \mathbf{false}.z\}\}\vec{x}$$

and noting its domain of definition is the specified subspace. □

Example 61 (finite covers)**.** A *finite measurable cover* of a space X is a list $\vec{E} \in \mathbf{List}\,\mathcal{B}_X$ such that:

$$\llcorner X \lrcorner = \bigcup \vec{E} \coloneqq \mathbf{fold}\,\lambda\,\{\mathbf{Nil}.\emptyset; E :- F.E \cup F\}\vec{E}$$

We define the measurable function $-$*-cover* $: (0, \infty) \to \mathbf{FinCover}\,\mathbb{R}$, by setting ε*-cover* to:

$$\mathbf{let}\,n : \mathbb{N} = \left\lceil \tfrac{1}{\varepsilon} \right\rceil\ \mathbf{in}\ [-\infty, -\tfrac{1}{\varepsilon}) :- [\tfrac{1}{\varepsilon}, \infty] :- \left[[-\tfrac{i}{\varepsilon N^2}, -\tfrac{i-1}{\varepsilon N^2})\middle|1 \le i \le n^2\right] +\!\!+ \left[[\tfrac{i-1}{\varepsilon N^2}, \tfrac{i}{\varepsilon N^2})\middle|1 \le i \le n^2\right]$$

It selects, for every ε, a finite cover of subsets with 'diameter' bounded by ε, i.e. events E satisfying:

$$(E \subseteq [-\infty, -\tfrac{1}{\varepsilon}]) \vee (E \subseteq [\tfrac{1}{\varepsilon}, \infty]) \vee (E \subseteq [-\tfrac{1}{\varepsilon}, \tfrac{1}{\varepsilon}], \forall x, y \in E.\,|x - y| \le \varepsilon)$$

Since every element in $\overline{\mathbb{R}}$ must belong to one of the events in a cover, the pigeon-hole principle implies that, for all $\vec{x} \in \overline{\mathbb{R}}^{\mathbb{N}}$ and $\varepsilon \in (0, \infty)$, there is an event E in ε*-cover* such that $\vec{x} \in E$-i.o.. So:

$$\lambda\,(\varepsilon, \vec{x})\,.\text{Ƨ}(\varepsilon\text{-}cover, \lambda E.\vec{x} \in E\text{-i.o.}) : (0, \infty) \times \overline{\mathbb{R}}^{\mathbb{N}} \to \mathcal{B}_{\overline{\mathbb{R}}}$$

selects for a subset of diameter bounded by ε that meets the sequence infinitely often. □

Inductive types can represent programs, and give a simplified example from Janz et al. [2016].

Example 62 (probabilistic context free grammars)**.** Consider the abstract syntax of probabilistic context free grammars (PCFGs, on left) with production rules in a space R and literals in X:

$$t \coloneqq t_1 \langle p \| 1 - p \rangle t_2 | t_1 ⨟ t_2 | \epsilon | r \in R | x \in X$$

We represent PCFGs as the inductive type satisfying the fix-point equation:

$$\begin{aligned}\mathbf{PCFG}\,R\,X = \{&\mathsf{toss} : (\mathbf{PCFG}\,R\,X, (0,1), \mathbf{PCFG}\,R\,X)\,, \\ &\mathsf{seq} : (\mathbf{PCFG}\,R\,X, \mathbf{PCFG}\,R\,X)\,, \mathsf{empty}, \mathsf{Rule} : R, \mathsf{Terminal} : \mathrm{X}\}\end{aligned}$$

In detail:

- $\mathsf{toss}(t_1, p, t_2)$ represents $t_1 \langle p \| 1 - p \rangle t_2$, a probabilistic alternation with bias p to the left subtree t_1 over the right subtree t_2.
- $\mathsf{seq}(t_1, t_2)$ represents the concatenation $t_1 ⨟ t_2$ of production terms t_1 and t_2; and
- empty represents the empty string terminal ϵ; and
- $\mathsf{Rule}\,r$ represent a (recursive) call to another production rule $r \in R$;
- $\mathsf{Terminal}\,x$ represents a terminal $x \in X$.

For simplicity, we will use the more abstract PCFG description, but the translation to the lower-level tree representation is straightforward.

For example, we define rules for producing uniformly distributed bits and bit-strings of a given length, i.e., the production rules are $R := \{\mathsf{One}, \mathsf{Many} : \mathbb{N}\}$

$$\mathrm{rule} : R \to \mathbf{PCFG}\, R\, \mathbf{Bit} \qquad \mathrm{rule} = \lambda \begin{Bmatrix} \mathsf{One}.0\langle \tfrac{1}{2} \| \tfrac{1}{2} \rangle 1 \\ \mathsf{Many}\, n.\mathbf{fold}\, \lambda \{0.\varepsilon; \mathsf{S}t.\, (\mathsf{One} ⨟ t)\}n \end{Bmatrix}$$

The first rule production is rule One is $0\langle \frac{1}{2} \| \frac{1}{2} \rangle 1$, the PCFG that would generate a uniformly distributed random bits. The second production rule, rule $(\mathsf{Many}\, n)$, produces a uniformly distributed sequence of n bits. Here we use the structural induction construct **fold** to recurse over the given natural number. In the base case, we will return the empty production. In the inductive case, given the result t for the recursive call, we concatenate the rule for one bit, via a (recursive) rule production to One with the result t. We will not revisit spaces of syntax further. □

5.3 Coinductive datatypes by example

We will demonstrate two versatile coinductive datatypes: streams and infinite trees.

Example 63 (streams)**.** The coinductive type $\mathbf{Stream}\, A$ of streams with elements in A is the function space $A^{\mathbb{N}}$. Its codata destructors project out the value stored in the head of the stream together with its tail stream:

$$\begin{array}{ll} (.\mathsf{head}) : \mathbf{Stream}\, A \to A & (.\mathsf{tail}) : \mathbf{Stream}\, A \to \mathbf{Stream}\, A \\ \vec{u}.\mathsf{head} := \vec{u}\, 0 & \vec{u}.\mathsf{tail} := \lambda n.\vec{u}(n+1) \end{array}$$

For example, we can always split off the first n elements in a stream:

$$\begin{array}{ll} \mathbf{split} : \mathbb{N} \times \mathbf{Stream}\, A \to \mathbf{List}\, A \times \mathbf{Stream}\, A & \mathbf{take} : \mathbb{N} \times \mathbf{Stream}\, A \to \mathbf{Stream}\, A \\ \mathbf{split}\, (n, \vec{u}) := \mathbb{N}.\mathbf{fold} & \mathbf{take}\, (n, \vec{x}) := (\mathbf{split}\, (n, \vec{x})).1 \\ \quad \lambda \begin{Bmatrix} 0.\lambda \vec{v}.\, (\mathbf{Nil}, \vec{v})\, ; \\ 1 + f.\lambda \vec{v}.\mathbf{let}\ (\vec{x}, \vec{w}) = f(\vec{v}.\mathsf{tail}) \\ \qquad \mathbf{in}\ (\vec{v}.\mathsf{head} :- \vec{x}, \vec{w}) \end{Bmatrix} n\, \vec{u} & \begin{array}{l} \mathbf{drop} : \mathbb{N} \times \mathbf{Stream}\, A \to \mathbf{Stream}\, A \\ \mathbf{drop}(n, \vec{x}) := (\mathbf{split}\, (n, \vec{x})).2 \end{array} \end{array}$$

By induction over the number n of elements to split off of a given stream $\vec{v}$. In the base case, we split off no elements, returning the empty list and the stream unchanged: $(\mathbf{Nil}, \vec{v})$. In the inductive case, we split the tail of the given stream into a list $\vec{x}$ and a stream $\vec{w}$, and place the given stream's head at the head of the split list: $(\vec{v}.\mathsf{head} :- \vec{x}, \vec{w})$. The functions **take** and **drop** are the corresponding projections of these resulting lists and stream.

To demonstrate coinduction, we define zip $: \mathbf{Stream}\, A \times \mathbf{Stream}\, B \to \mathbf{Stream}\, (A \times B)$:

$$\mathrm{zip}\, (\vec{x}, \vec{y}) := \mathbf{Stream.unfold}\, \lambda(\vec{u}, \vec{v}).\mathbf{<}\mathsf{head} : (\vec{u}.\mathsf{head}, \vec{v}.\mathsf{head}), \mathsf{tail} : (\vec{u}.\mathsf{tail}, \vec{v}.\ell tail)\mathbf{>}(\vec{x}, \vec{y})$$

The coinduction operation, **unfold**, takes: a function mapping a *seed* to the next element in the stream and a new seed; and an initial seed. The result is then the stream of successive elements this function generates starting with the initial seed. In this case, at each iteration we project out the two heads for the two streams, and proceed with their tails as seeds. □

Example 64 (subsequence selection)**.** We define a principle for selecting, given an event and a sequence that satisfies the event infinitely often, a subsequence satisfying the event:

$$\mathbf{subseq} : \{(\vec{x}, E) \in \mathbf{Stream}\, X \times \mathcal{B}_X | \vec{x} \in E\text{-i.o.}\} \to \{\vec{y} \in \mathbf{Stream}\, X | \forall n.y_n \in E\}$$

Let $D := \{(\vec{x}, E) \in \mathbf{Stream}\, X \times \mathcal{B}_X | \vec{x} \in E\text{-i.o.}\}$. We take D as our type of seeds, and define **subseq** by iteratively selecting a next element in E:

$$\mathbf{subseq}\,(\vec{x}, E) = \mathbf{Stream.unfold}\,\lambda\,(\vec{y}, F) : D.(\mathbf{let}\ n = \text{Ƨ}(\vec{y}, F)\ \mathbf{in} <\mathsf{head} : y_n; \mathsf{tail} : (\mathbf{drop}(1 + n, \vec{y}), F)>)(\vec{x}, E)$$

We used the general sequential selection principle from Example 58. Using the stronger selection guarantee for Ƨ, we can guarantee that the resulting sequence will contain *all* elements in E:

$$\mathbf{subseq} : \{(\vec{x}, E) | \vec{x} \in E\text{-i.o.}\} \to \{\vec{y} | (\forall n.y_n \in E), (\forall m.x_m \in E \Rightarrow \exists n.x_m = y_m)\}$$ □

Example 65 (rational approximation revisited, cf. Example 10)**.** We can define the rational approximation function as in the introduction since an enumeration of the rationals $\vec{q} : \mathbb{N} \twoheadrightarrow \mathbb{Q}$ meets every non-empty interval infinitely often:

$$\mathsf{approx} : \mathbb{R} \to \mathbb{Q}^{\mathbb{N}} \qquad \mathsf{approx}\, r := \lambda n.\text{Ƨ}(\vec{q}, (r - \tfrac{1}{n}, r))$$

The function $\mathsf{approxFun} : \mathbb{R}^{\mathbb{R}} \to (\mathbb{Q}^{\mathbb{R}})^{\mathbb{N}}$ from the introduction is measurable. □

Example 66 (subsequences)**.** A *subsequence* is a strictly increasing sequence of indices, and we form the subspace $\mathbf{Subseq} \hookrightarrow \mathbb{N}^{\mathbb{N}}$ by taking $\mathbf{Subseq} := \left\{\vec{k} \middle| \forall n.k_n < k_{n+1}\right\}$. The embedding is Borel because its defining predicate is measurable by type-checking. Subsequences are also coinductive, and their coinduction coincides with stream coinduction, since the two spaces are isomorphic:

$$\varphi : \mathbf{Subseq} \xrightarrow{\cong} \mathbf{Stream}\,\mathbb{N} \qquad \varphi\vec{k} := \lambda\,\{0.k_0; 1 + n.k_{n+1} - k_n - 1\}$$

To avoid confusion, we will not use subsequence coinduction, only stream coinduction.

We can strengthen the selection principle from Example 64 to select a subsequence:

$$\mathbf{subseq} : \{(\vec{x}, E) | \vec{x} \in E\text{-i.o.}\} \to \left\{\vec{k} \in \mathbf{Subseq} \middle| \forall n.x_{k_n} \in E\right\}$$

Do so, note that the following function lifts to a function between the two subspaces:

$$\lambda(\vec{k}, E).(\mathrm{zip}\,(\vec{k}, (n)_n), E) : \{(\vec{x}, E) | \vec{x} \in \text{-i.o.}\} \to \left\{(\vec{u}, E) \middle| \vec{u} \in \pi_1^{-1}[E]\text{-i.o.}\right\}$$

We can therefore apply the selection principle from Example 64 and project out its index, which gives us the subsequence we need:

$$\mathbf{subseq}(\vec{k}, E) := \mathbf{let}\ \vec{w} = \mathbf{subseq}(\mathrm{zip}\,(\vec{k}, (n)_n), E)\ \ \mathbf{in}\ \ \mathbf{Stream.map}\,\pi_2\vec{w}$$

We use the stronger selection principle to select, given a sequence of events that meet a sequence of points infinitely often, a subsequence where each point satisfies the corresponding event:

$$\mathbf{subseq} : \limsup_{n\to\infty} \left\{(\vec{x}, \vec{E}) \middle| x_n \in E_n\right\} \to \left\{\vec{k} \in \mathbf{Subseq} \middle| \forall n.x_{k_n} \in E_{k_n}\right\}$$

Let $\mathcal{E} : \mathcal{B}_{X \times \mathcal{B}_X}$ be the set $\{(y, F) | y \in F\}$. Then $\mathrm{zip}\,(\vec{x}, \vec{E}) \in \mathcal{E}$-i.o. in the domain of definition, and so we may select the subsequence by $\mathbf{subseq}(\vec{x}, \vec{E}) := \mathbf{subseq}(\mathrm{zip}\,(\vec{x}, \vec{E}), \mathcal{E})$. □

Example 67 (universal subsequence selection)**.** We will sometimes need an even stronger subsequence selection principle, namely that is selects the maximal, or universal, subsequence. Formally, we define a partial order on subsequences $\vec{k} \leq \vec{\ell}$ when there exists a subsequence $\vec{m}$ that exhibits $\vec{\ell}$ as a subsequence of $\vec{k}$: $\forall n.k_{m_n} = \ell_n$. We have:

$$\mathbf{subseq} : \{(\vec{x}, E) | \vec{x} \in E\text{-i.o.}\} \to \left\{\vec{k} \in \mathbf{Subseq} \middle| (\forall n.x_{k_n} \in E), (\forall \vec{\ell} \in \mathbf{Subseq}.(\forall n.x_{\ell_n} \in E) \Rightarrow \vec{k} \leq \vec{\ell})\right\}$$

The selection principle we have defined already satisfies this specification, since it selects *all* the positions for which $x_n \in E$. □

Example 68 (partial limits)**.** Let $\vec{x} \in \mathbf{Stream}\,\overline{\mathbb{R}}$ be a sequence. Recall that $L \in \overline{\mathbb{R}}$ is a *partial limit* for $\vec{x}$ when there is a subsequence of $\vec{x}$ that converges to L. We can select such a subsequence. First, define the following auxiliary function that yields a sequence of events that guarantee a sequence converges to a given limit:

$$\lambda L.\vec{E}^L : \left\{L \in \overline{\mathbb{R}}\right\} \to \left\{\vec{E} \in \mathcal{B}_{\overline{\mathbb{R}}}^{\mathbb{N}} \middle| \forall \vec{x} \in \overline{\mathbb{R}}^{\mathbb{N}}.(\forall n.\vec{x} \in E^L_n\text{-eventually}) \iff L = \lim_{n\to\infty} x_n\right\}$$
$$\vec{E}^L \coloneqq \mathbf{case}\ L\ \mathbf{of}\ \{\infty.\ ([n,\infty])_n\,; -\infty.\ ([-\infty,-n])_n\,; \text{otherwise}.\mathbb{R} :- \left((L - \tfrac{1}{n}, L + \tfrac{1}{n})\right)_n \}$$

To select the sub-sequence, take the type of seeds to be:

$$D \coloneqq \left\{\left(\vec{x}, L, \vec{E}, \vec{k}\right) \in \mathbf{Stream}\,\overline{\mathbb{R}} \times \overline{\mathbb{R}} \times \mathbf{Stream}\,\mathcal{B} \times \mathbf{Subseq} \middle| \exists \vec{\ell}.x_{k_{\ell_n}} \to L\right\}$$

and define **subseq** by iterating through $\vec{E}^L$ and iteratively selecting a subsequence in each E^L_n:

$$\begin{aligned}
&\mathbf{subseq} : \left\{(\vec{x}, L) \in \mathbf{Stream}\,\overline{\mathbb{R}} \times \overline{\mathbb{R}} \middle| \exists \vec{\ell}.x_{k_{\ell_n}} \to L\right\} \to \left\{\vec{k} \in \mathbf{Subseq} \middle| \lim_n x_{k_n} = L\right\}\\
&\mathbf{subseq}(\vec{x}, L) \coloneqq \mathbf{Stream}\,.\mathbf{unfold}\,(\lambda(\vec{x}, L, \vec{E}, \vec{k}) \in D.\mathbf{let}\ \vec{\ell} = \text{Ƨ}(\left(x_{k_n}\right)_n, E.\mathsf{head})\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad \mathbf{in} <\mathsf{head} : k_{\ell_0}, \mathsf{tail} : (\vec{x}, L, \vec{E}.\mathsf{tail}, \left(k_{\ell_{n+1}}\right)_n)>)\\
&\qquad\qquad\qquad\qquad\qquad\qquad (\vec{x}, L, \vec{E}^L, (n)_n)
\end{aligned}$$

Let $\vec{m} = \mathbf{subseq}(\vec{x}, L)$. By induction on n, the seed for iteration n also satisfies, for all $i < n$, $x_{k_m} \in E^L_i$. As a consequence, $x_{m_\ell} \in E^L_n$ for all $\ell \geq n$. So $\forall n.\ \left(x_{m_n}\right)_n \in E^L_n$-eventually, and so $x_{m_n} \to L$. □

Example 69 (Bolzano-Weierstrass)**.** We can select a converging subsequence for every sequence in $\overline{\mathbb{R}}^{\mathbb{N}}$. The existence of such a sequence is the Bolzano-Weierstrass (BW) theorem for $\overline{\mathbb{R}}$, and we will show two selection principles for it: $\mathbf{BW} : \left\{\vec{x} \in \mathbf{Stream}\,\overline{\mathbb{R}}\right\} \to \left\{\vec{k} \in \mathbf{Subseq} \middle| \exists L \in \overline{\mathbb{R}}.L = \lim_n x_{k_n}\right\}$.

We can select this subsequence by assuming the BW theorem. If we know that there is a subsequence with a limit, we know that there is a sub-sequence whose limit is $L \coloneqq \limsup_n x_n$, which we can select. We then use the previous example: $\mathbf{BW}\,\vec{x} \coloneqq \mathbf{subseq}(\vec{x}, \limsup \vec{x})$. We can however turn the proof of the BW theorem into a selection principle, and along the way demonstrate a more sophisticated use of induction and coinduction:

$$\begin{aligned}
&\mathbf{BW}\,\vec{x} \coloneqq \mathbf{if}\ \nexists N \in \mathbb{N}.\forall n.x_n \leq +N\ \mathbf{then}\ \mathbf{subseq}(\vec{x}, +\infty)\\
&\qquad \mathbf{else\ if}\ \nexists N \in \mathbb{N}.\forall n.x_n \geq -N\ \mathbf{then}\ \mathbf{subseq}(\vec{x}, -\infty)\\
&\qquad \mathbf{else\ let}\ N = \text{Ƨ}(\lambda N.\forall n.\ |x_n| < N)\\
&\qquad\quad \mathbf{in\ unfold}\,(\lambda(\vec{x}, N, \varepsilon, \vec{k}) \in \left\{(\vec{x}, N, \varepsilon, \vec{k}) \middle| N \leq \tfrac{1}{\varepsilon}, \forall n.\ \left|x_{k_n}\right| < N, \forall n, m.\ \left|x_{k_n} - x_{k_m}\right| \leq \varepsilon\right\}.\\
&\qquad\qquad\quad \mathbf{let}\ E = \text{Ƨ}(\tfrac{\varepsilon}{2}\text{-}\mathit{cover}, \lambda E.\ \left(x_{k_n}\right)_n \in E\text{-i.o.})\\
&\qquad\qquad\qquad\ \vec{\ell} = \mathbf{subseq}(\left(x_{k_n}\right)_n, E)\\
&\qquad\qquad\quad \mathbf{in} <\mathsf{head} : k_{\ell_0}; \mathsf{tail} : (\vec{x}, \tfrac{\varepsilon}{2}, \left(k_{\ell_n}\right)_n)>)\\
&\qquad\qquad\quad (\vec{x}, N, \tfrac{1}{N}, (n)_n)
\end{aligned}$$

First, we select whether the sequence is contained in some some bounded interval. If it is not, it has $\pm\infty$ as a limit superior-inferior and meets $E^{\pm\infty}$ infinitely-often, and we can select a corresponding subsequence. Otherwise, select a—necessarily positive—natural number $N \in \mathbb{N}$ such that $(-N, N)$ bounds the sequence.

We use coinduction, seeding the sequence in question and a subsequence of it whose elements are bounded by N and are at most ε apart from each other for $N \leq \frac{1}{\varepsilon}$, starting with the initial seed given by the selected bound and it reciprocal, and the full sequence: $(\vec{x}, N, \frac{1}{N}, (n)_n)$. At each coinductive step, select one of the sets E from the finite measurable cover $\frac{\varepsilon}{2}$-*cover* whose diameter is at most $\frac{\varepsilon}{2}$ so that E meets the given subsequence infinitely often (cf. Example 61). Since $\left|x_{k_n}\right| <$

$N \leq \frac{1}{\varepsilon}$, it must be the case that $E \subseteq [-\frac{1}{\varepsilon}, \frac{1}{\varepsilon}]$, and its diameter is $\frac{\varepsilon}{2}$. We then select a subsequence $\vec{\ell}$ from E. The elements of this subsequence are at most $\frac{\varepsilon}{2}$ apart. The current index is then the first index of the subsequence k_{ℓ_0}, and we seed the next iteration with half the diameter and the rest of the subsequence. Let $\vec{k}$ be the subsequence resulting from this coinduction. It is a Cauchy sequence by induction on n, as $\forall m, \ell \geq n$ we have $\left|x_{k_m} - x_{k_\ell}\right| \leq \varepsilon$. □

So far we studied only induction and coinduction that recurses on one substructure. We demonstrate the more general induction and coinduction using well-founded and infinite trees.

Example 70 (infinite trees, Cantor schemes)**.** The type **InfiniTree** A of infinite binary trees with nodes taken from A is the function space $A^{\mathbf{List\,Bit}}$. Its codata destructors project out the value stored in the root node its two subtrees:

$$\begin{array}{ll} (.\mathsf{val}) : \mathbf{InfiniTree}\,A \to A & (.0), (.1) : \mathbf{InfiniTree}\,A \to \mathbf{InfiniTree}\,A \\ t.\mathsf{val} := t\,[0] & t.b := \lambda\vec{b}.t(b :- \vec{b}) \end{array}$$

While we can define trees directly, we can also define them coinductively, explicitly, and measurably, passing a seed parameter $s : S$, generating the current value and seeds for each subtree:

$$\mathbf{InfiniTree}\,A.\mathbf{unfold} \; : (S \to \lhd \mathsf{val} : A, 0 : S, 1 : S \rhd) \to \mathbf{InfiniTree}\,A$$

A *Cantor scheme* over X is a infinite tree over $\mathcal{B}_X$, where each sibling nodes are disjoint and contained in their parent node, letting **CantorScheme** $X \hookrightarrow$ **InfiniTree** $\mathcal{B}_X$ be the induced subspace:

$$t \in \mathbf{CantorScheme}\,X \iff \forall \vec{b} \in \mathbf{List\,Bit}.(t(\vec{b} +\!\!+ [0]) \cap t(\vec{b} +\!\!+ [1]) = \emptyset) \quad , \quad \forall b \in \mathbf{Bit}.t(\vec{b} +\!\!+ [b]) \subseteq t\vec{b}$$

Every Cantor scheme t encodes a measurable set $(\!|t|\!)$ and this encoding is measurable:

$$(\!|-|\!) : \mathbf{CantorScheme}\,X \to \mathcal{B}_X \qquad (\!|t|\!) := \bigcap\nolimits_{n \in \mathbb{N}} \bigcup\nolimits_{\vec{b} \in \mathbf{List\,Bit}, \mathbf{length}\,\vec{b} = n} t\vec{b}$$

We define the well-known *Cantor set*, by coinductively removing middle thirds in the seed:

$$\mathbb{G} := (\!|\mathbf{InfiniTree.unfold}\,(\lambda(a, b) \in \{(a, b) | a < b\}\,.\!<\!\mathsf{val} : [a, b], 0 : (a, \tfrac{b-a}{3}), 1 : (2\tfrac{b-a}{3}, b)\!>\!)(0, 1)|\!)$$

While it is not difficult to define the Canto set directly, with coinduction it is succinct. □

Example 71. One application for Cantor schemes is to define the following bijection between the interval $[0, 1]$ and the the *Cantor space* **Stream Bit**. Consider the Cantor scheme with $[\![t]\!] = [0, 1)$:

$$t := \mathbf{unfold}\,(\lambda\,(a, b) \in \{(a, b) | a < b\}\,.\!<\!\mathsf{val} : [a, b), 0 : (a, \tfrac{a+b}{2}), 1 : (\tfrac{a+b}{2}, b)\!>\!)(0, 1)$$

We can select an inhabitant for each node in t, for example:

$$(\!|-|\!) : \left\{\vec{b} \in \mathbf{List\,Bit}\right\} \to \left\{x \in [0, 1) \middle| x \in t\vec{b}\right\} \qquad (\!|[b_1, \ldots, b_n]|\!) := \sum_{i=1}^{n} \frac{b_i}{2^n}$$

Let $D \hookrightarrow [0, 1]$ be the space positive diadic numbers $D := \left\{\frac{2m+1}{2^n} \middle| m, n \in \mathbb{N}, 2m + 1 \leq 2^n\right\}$. Each number $x \in [0, 1] \setminus D$ has a unique binary expansion $\vec{b}$, and for each prefix $\vec{b}\Big|_n := [b_0, \ldots, b_{n-1}]$, we have that $x \in t\;\vec{b}\Big|_n$, and so $x \in (\!|t|\!)$. The diadic numbers in D generate edge cases: each $1 \neq x \in D$ has a trailing-0 and trailing-1 representation, and only the the trailing-0 representation satisfies the property $x \in t\;\vec{b}\Big|_n$. Moreover, the diadic $1 \in D$ does not have such a trailing-0 representation, and $1 \notin (\!|t|\!)$. The crux of the construction is the following isomorphism, that uses the Cantor scheme:

$$\varphi : \left\{\vec{b} \in \middle| \vec{b} \in \{0\}\text{ -i.o.}\right\} \xrightarrow{\cong} (\!|t|\!) = [0, 1) \qquad \varphi\vec{b} := \lim\nolimits_{n \to \infty} (\!|\vec{b}\Big|_n|\!) \qquad \varphi^{-1}x := \lambda\vec{b}.[x \in t(\vec{b} -: 1)]$$

$$
\begin{array}{llll}
\mathbf{Type} \ni A, B \coloneqq & \cdots & & \text{types} \\
\mid & F(A_1, \ldots, A_n) & & \text{type family} \\
\mid & \boldsymbol{\mu}\vec{\alpha}.\boldsymbol{\nu}\vec{\beta}.F & & \text{bi-inductive type}
\end{array}
\qquad
\begin{array}{llll}
\mathbf{preTerm} \ni M, N \coloneqq & \cdots & & \text{terms} \\
\mid & F.\mathbf{map}\,M & & \text{map} \\
\mid & F.\mathbf{bifold}\,M & & \text{bi-induction}
\end{array}
$$

Fig. 6. Mixed inductive and coinductive datatypes and mixed induction and coinduction

I.e., we map each bit sequence to the number whose binary expansion it represents, and conversely, the Cantor scheme reconstructs the binary expansion of the number. Next, let $D' \coloneqq D \setminus \{1\}$. Then $⦇-⦈$ is an isomorphism between non-empty bit sequences and D'. Consequently, we have a bijection $\xi : D' \amalg D' \xrightarrow{\cong} D'$ for example, by prepending a 0 or 1 depending on the tag.

$$
\psi : \mathbf{Stream}\,\mathbf{Bit} \xrightarrow{\cong} [0,1) \amalg D \qquad \psi \coloneqq \lambda \left\{\begin{array}{ll} [1, 1, \ldots] & .\iota_2 1 \\ [b_0, \ldots, b_{n-1}, 0, 1, 1, \ldots] & .\iota_2 ⦇[b_0, \ldots, b_{n-1}, 1]⦈ \\ \vec{b} & .\iota_1 \varphi \vec{b} \end{array}\right\}
$$

Putting everything together:

$$
\mathbf{Stream}\,\mathbf{Bit} \xrightarrow{\psi} [0,1) \amalg D \cong ([0,1] \setminus D') \amalg (D' \amalg D') \xrightarrow{\mathrm{id} \amalg \xi} ([0,1] \setminus D') \amalg D' \cong [0,1]
$$

And this construction is the desired isomorphism. □

5.4 Datatypes, induction and coinduction

These examples motivate the use of a general type-former that gives us access to inductive and coinductive datatypes and induction and coinduction over them. The formal treatment of datatypes is technical, but standard, using initial algebra semantics Goguen et al. [1977]. Unfortuantely, we did not prepare the appendix for them in time, but we will recount them in Appendix A in the final submission. Here we demonstrate how to reduce the last few examples to a single construct.

Fig. 6 presents the constructs allowing us to manipulate inductive and coinductive types. To support type families such as $\mathbf{List}\,A$, the type system in fact supports type families F that include type variables such as $\alpha, \beta \vdash \lhd \mathsf{head} : \alpha, \mathsf{tail} : \beta \rhd$. We can then (partially or fully) instantiate these type families $F(A_1, \ldots, A_n)$ with a tuple of types via substitution, e.g., in the previous example: $\lhd \mathsf{head} : \alpha, \mathsf{tail} : \beta \rhd(\mathbb{R}, \mathbb{N}) = \lhd \mathsf{head} : \mathbb{R}, \mathsf{tail} : \mathbb{N} \rhd$. We can uniformly apply a function to a family using the map operation: $\Gamma \vdash F.\mathbf{map}\ M : F(\vec{A}) \to F(\vec{B})$. We can take the *mixed* inductive fixpoint of a family $\boldsymbol{\mu}\vec{\alpha}.\boldsymbol{\nu}\vec{\beta}.F$ with the **bifold** construct:

$$
\frac{\Gamma \vdash M : \lhd \mathsf{alg} : \boldsymbol{\mu}\vec{\alpha}.F(\vec{\alpha}, \vec{B}) \to F(\vec{A}, \vec{A}[2]), \mathsf{coalg} : F(\vec{A}, \vec{B}) \to \boldsymbol{\nu}\vec{\beta}.F(\vec{A}, \vec{\beta}) \rhd}{\Gamma \vdash F.\mathbf{bifold}\ M : \boldsymbol{\mu}\vec{\alpha}.F(\vec{\alpha}, \vec{B}) \to \boldsymbol{\nu}\vec{\beta}.F(\vec{A}, \vec{\beta})}
$$

Example 72. We recover our previous examples for inductive and coinductive types by:

$$
\mathbb{N} = \boldsymbol{\mu}\alpha.\{0, (1+) : \alpha\} \qquad \mathbf{List}\,\beta = \boldsymbol{\mu}\alpha.\{\mathsf{Nil}, (\,:\,) : (\beta, \alpha)\} \qquad \mathbf{Stream}\,\beta = \boldsymbol{\nu}\alpha.\lhd \mathsf{head} : \beta, \mathsf{tail} : \alpha \rhd
$$
$$
\mathbf{InfiniTree}\,\beta = \boldsymbol{\nu}\alpha.\lhd \mathsf{val} : \alpha, 0 : \alpha, 1 : \alpha \rhd
$$
□

6 CONCRETE SPACES

Concrete spaces, such at the discrete countable spaces $\mathbb{B}$ and $\mathbb{N}$, and the Euclidean spaces $\mathbb{R}^{\mathbb{N}}$ have a well-behaved theory of measurability. We define the concrete spaces like in the classical case.

6.1 Standard spaces and their basic properties

A space S is *standard* when there is a Borel subset $E \in \mathcal{B}$ and measurable isomorphism $\varphi : S \xrightarrow{\cong} E$. A space S is standard iff there is a Borel embedding $S \hookrightarrow \mathbb{R}$, and iff there is a partial-map section $S \xrightarrow{e} \mathbb{R} \xrightarrow{p} S$ as in Example 53.

By Example 38, the metaphorology and the σ-field of the reals determine each other uniquely:

$$\overset{\mathbf{Meas}}{\ulcorner \mathbb{R} \urcorner} = \mathbb{R} \qquad \underset{\mathbf{Qbs}}{\llcorner \mathbb{R} \lrcorner} =: \mathbb{R}$$

and so, by Example 46, a function into or out of the reals is measurable iff it is quasi-measurable. This situation extends to every Borel subspace i.e.: for every standard space $S \hookrightarrow \mathbb{R}$, we have $\mathcal{R}_{\llcorner \ulcorner S \urcorner \lrcorner} = \mathcal{R}_S$, and for every standard Borel space, i.e., a measurable space S that is isomorphic to a Borel subset, we have $\mathcal{B}_{\ulcorner \llcorner S \lrcorner \urcorner} = \mathcal{B}_S$.

Indeed lets treat quasi-Borel spaces first. Assume S is standard space, i.e., there is an embedding $e : S \hookrightarrow \mathbb{R}$. We will show that $\underset{\mathbf{Qbs}}{\llcorner \overset{\mathbf{Meas}}{\ulcorner S \urcorner} \lrcorner}$ and S have the same metaphorologies. For every space X, $\mathcal{R}_X \subseteq \mathcal{R}_{\llcorner \ulcorner X \urcorner \lrcorner}$, for if $\alpha \in \mathcal{R}_X$, then $\alpha : \ulcorner \mathbb{R} \urcorner \to \ulcorner X \urcorner$, and so $\alpha : \mathbb{R} \to \llcorner \ulcorner X \urcorner \lrcorner$, i.e. $\alpha \in \mathcal{R}_{\llcorner \ulcorner X \urcorner \lrcorner}$. Conversely, take $\alpha \in \mathcal{R}_{\llcorner \ulcorner S \urcorner \lrcorner}$. If $E \in \mathcal{B}_{\mathbb{R}}$, then, by Example 47, $e^{-1}[E] \in \mathcal{B}_S = \mathcal{B}_{\ulcorner S \urcorner}$, and so $(e \circ \alpha)^{-1}[E] = \alpha^{-1}[e^{-1}[E]] \in \mathcal{B}_{\mathbb{R}}$. Therefore $e \circ \alpha \in \mathcal{R}_{\mathbb{R}}$ and since $S \hookrightarrow \mathbb{R}$ we deduce $\alpha \in \mathcal{R}_S$. Therefore the two metaphorologies agree.

Lets treat measurable spaces. Take any standard Borel space S, and we will show $\mathcal{B}_{\ulcorner \llcorner S \lrcorner \urcorner} = \mathcal{B}_S$. For every measurable space A, we have $\mathcal{B}_A \subseteq \mathcal{B}_{\ulcorner \llcorner A \lrcorner \urcorner}$ by Example 46, since $\lambda x.x : \llcorner A \lrcorner \to \llcorner A \lrcorner$ is measurable. Conversely, take $E \in \mathcal{B}_{\ulcorner \llcorner A \lrcorner \urcorner}$. If $E = \llcorner S \lrcorner$, then $E \in \mathcal{B}_A$. Otherwise, there is some $a \notin E$. Letting $e : S \hookrightarrow \mathbb{R}$ be the embedding of the measurable space as a Borel subspace, we then have a random element $\alpha : \mathbb{R} \to S$, $\alpha = \left[e[S].e^{-1}, (e[S])^{\complement}.\underline{a}\right]$. We therefore have $\mathcal{B}_{\mathbb{R}} \ni \alpha^{-1}[E] = e[E]$. Therefore $e[E]$ is Borel in $\mathbb{R}$, so $e[E]$ is Borel in $e[S]$ and E is Borel in S.

As a consequence, for standard spaces, whether considered as measurable or quasi-Borel, the two putative notion of measurability agree. We can therefore transfer classical results concerning standard Borel spaces to corresponding results concerning standard spaces. We will, however, not do so. Instead, we will demonstrate how to use our spaces to derive classical results about standard Borel spaces. When possible, we take full advantage of the better type-structure these spaces support. Otherwise, the classical proofs transfer verbatim. For example:

Example 73. The following standard spaces are isomorphic to each other: $\mathbb{R}$, $\overline{\mathbb{R}}$, $[a, b]$, $[a, b)$, $(a, b]$, for all $a < b$ in $\overline{\mathbb{R}}$. Indeed, $\lambda x.\frac{1}{x} : [0, 1] \xrightarrow{\cong} [0, \infty]$, restricting to $[0, 1) \cong (0, \infty]$, $(0, 1] \cong [0, \infty)$ and $(0, \infty) \cong (0, 1)$. For $-\infty < a, b < \infty$, we have $\lambda x.a + (b - a)x : [0, 1] \xrightarrow{\cong} [a, b]$. We also have: $\lambda \left\{\frac{1}{n}.\frac{1}{n+1}; x.x\right\} : [0, \infty] \xrightarrow{\cong} [0, \infty)$ and since this isomorphism sends ∞ to 0, it restricts to an isomorphism $[0, \infty) \cong (0, \infty)$. The rest follow from these and affine transformations. $\square$

Example 74 (countable products)**.** The standard spaces are closed under countable products. We will start with $\mathbb{R}$. Take a countable I. If I is empty, $\mathbb{R}^I = \mathbb{1} \cong \{0\}$ is standard. Otherwise, we have $I \times \mathbb{N} \cong \mathbb{N}$. Note that $I \times \mathbb{N}$ is discrete since its random elements are σ-simple. Indeed, if $\alpha = \left[E_i.\underline{i}\right] \in \mathcal{R}_I$ and $\beta = \left[F_n.\underline{n}\right] \in \mathcal{R}_{\mathbb{N}}$ are σ-simple, then $\lambda r.(\alpha\, r, \beta\, r) = \left[E_i \cap F_n.(i, n)\right]_{(i,n)}$, i.e., a σ-simple function. Hence the bijection $I \times \mathbb{N} \cong \mathbb{N}$ is measurable, giving the chain of isomorphisms:

$$\mathbb{R}^I \cong [0, 1]^I \cong \mathbf{Stream\,Bit}^I \cong \mathbf{Bit}^{I \times \mathbb{N}} \cong \mathbf{Bit}^{\mathbb{N}} \cong \mathbb{R}$$

Therefore $\mathbb{R}^I$ is standard. In general, if $e_i : S_i \hookrightarrow \mathbb{R}$ are standard spaces, then their Cartesian product embeds as $\prod_i e_i : \prod_i S_i \hookrightarrow \mathbb{R}^I$, since: a product of injections in injective, and this product reflects random elements. Its image is the box $\prod_i e_i[S_i] = \bigcap_{i\in I} \pi_i^{-1}[e_i[S_i]]$, and so it is a Borel embedding, and $\prod_i S_i$ is standard. □

Example 75 (countable coproducts)**.** The standard spaces are also closed under countable coproducts. First, note that $\coprod_{i\in I} \mathbb{R} \cong I \times \mathbb{R}$, and this space is standard by Example 74. For general standard spaces $e_i : X_i \hookrightarrow \mathbb{R}$, their coproduct is also a Borel embedding $\coprod_i e_i : \coprod_i X_i \hookrightarrow \coprod_i \mathbb{R}$. □

Example 76 (subspaces and pullbacks)**.** A Borel subspace of standard space is standard since Borel embeddings are closed under composition. In particular, for a cospan $S \xrightarrow{f} U \xleftarrow{g} T$ between standard spaces, the pullback is standard. Indeed, $S \times T$ is standard by Example 74, and the pullback $f \bowtie g$ has a Boolean classifier, since the equality relation is measurable $(=) : U^2 \hookrightarrow \mathbb{R}^2 \to \mathbb{B}$. □

6.2 Compatible metrics and generating families

Since a standard space is completely determined by its σ-field, it is natural to find generating families of events for it. Moreover, as we wish to recover results from classical measure theory where events are primitive, we will develop some tools to this end. We will not directly use these tools in the remainder—only during proofs—and so disinterested readers can safely skip to §6.3.

Definition 3. *Let X be a space and $d : \llcorner X \lrcorner \to \overline{\mathbb{R}}$ be a metric over its points. The metric d is* semi-compatible *with X when, for every $x \in \llcorner X \lrcorner$, the function $\lambda y.d(x, y) : X \to \mathbb{W} := [0, \infty]$ is measurable. The metric d is* compatible *with X when it is a measurable function $d : X^2 \to [0, \infty]$.*

Example 77 (spaces underlying metric spaces)**.** Every metric is semi-compatible with the space underlying its metric space. Indeed, given a metric space $X = (\llcorner X \lrcorner, d)$ and $x \in \llcorner X \lrcorner$, we have $d(x, -)^{-1}[0, \varepsilon) = \mathrm{B}^d_\varepsilon x$ is the open ball around x of radius ε, hence Borel in the generated σ-field and in its underlying space. Thus $d(x, -)$ is measurable out of the induced measurable space, and, since $\mathbb{W}$ is standard Borel, then $d(x, -)$ is also measurable between the underlying spaces.

When the metric is separable, then it is also compatible with the underlying space underlying. Indeed, let B be countable and dense. Then

$$d^{-1}(a, \infty] = \bigcup_{b,c\in B} \{(x, y) | \exists \varepsilon_1, \varepsilon_2, \varepsilon_3 \in \mathbb{Q} \cap (0, \infty).d(b, x) < \varepsilon_1, d(c, y) \leq \varepsilon_2, d(b, c) > \varepsilon_3\}$$

and each subset in the countable union is Borel since d is semi-compatible. □

When a metric d is compatible with X, its induced σ-field consists of events: $\mathcal{B}_d \subseteq \mathcal{B}_X$. Moreover, the singleton function is measurable $\{-\} : X \to \mathcal{B}_X$, given by $\{x\} = d^{-1}(x, -)\,\{0\}$, and more generally for every countable I we have a measurable function $\{-\} : X^I \to \mathcal{B}_X$. As a consequence, the sequences converging to a given limit is an event:

$$\left\{(\vec{x}, x) \in X^{\mathbb{N}} \times X \middle| x = \lim_{n\to\infty} x_n\right\} = \left\{(\vec{x}, x) \middle| \forall \varepsilon \in \mathbb{Q}_+.\exists n.\forall m.m \geq n \implies x \in \mathrm{B}_\varepsilon x_m\right\} \in \mathcal{B}_X$$

Definition 4. *A metric d compatible with X has* measurable limits *when the partial function sending each sequence to its limit is partial* $\lim : X^{\mathbb{N}} \rightharpoonup X$*, i.e.:*

$$\mathbf{Converge}_d X := \left\{\vec{x} \in X^{\mathbb{N}} \middle| \exists \lim_n x_n\right\} \in \mathcal{B}_X \qquad \lim : \mathbf{Converge}_d X \to X$$

Example 78. The distance metric has measurable limits w.r.t. the Euclidean spaces. More generally, in every Cauchy-complete compatible metric the convergent sequences form an event as we can detect convergence via the Cauchy property. However, we still need to prove separately that the limit function is measurable over its domain of definition. □

Given a separable metric space X, an enumeration of a countable dense set $\vec{\mathbf{b}} : \mathbb{N} \twoheadrightarrow B$ lets us select a B-sequence for each point over *measurable* spaces:

$$\text{Ƨ} : \{x : X\} \to \left\{\vec{b} \in B^{\mathbb{N}} \middle| x = \lim_n b_n\right\} \text{ in } \mathbf{Meas} \qquad \text{Ƨ}x := \left(\text{Ƨ}(\vec{\mathbf{b}}, \lambda b.x \in \mathrm{B}_{\frac{1}{1+n}}\,b)\right)_n$$

The result to take from this subsection is a sufficient condition for a metric to generate all events:

Proposition 5. *Let d be a metric compatible with X that has measurable limits. If d is separable, then it generates all the events in this space: $\mathcal{B}_d = \mathcal{B}_X$.*

Proof

Since d is compatible, $\mathcal{B}_d \subseteq \mathcal{B}_X$. For the converse, take an event $E \in \mathcal{B}_X$. We then have the event:

$$E' := \left\{\vec{b} \in B^{\mathbb{N}} \middle| \vec{b} \in \mathbf{Converge}_d X \wedge \lim_n b_n \in E\right\} \in \mathcal{B}_{B^{\mathbb{N}}} \qquad x \in E \iff \text{Ƨ}x \in E'$$

Where E is an event by type-checking. Noting that $B^{\mathbb{N}}$ is standard, we can move to measurable spaces, and note that $E = \text{Ƨ}^{-1}[E'] \in \mathcal{B}_d$, as we wanted. ∎

6.3 Continuous-function space

While the full function space $\overline{\mathbb{R}}^{\overline{\mathbb{R}}}$ does not have a measurable space structure with a measurable evaluation function, some subspaces of it do. We study two such subspaces: in this subsection, the continuous-function subspace $\mathbf{C}_0 A$ of continuous functions $f : A \to \overline{\mathbb{R}}$ over a topological space A; and in Sec. 7, the space of functions that are right-continuous with left-limits at every point (càdlàg). For simplicity, we restrict to the case $A = I \subseteq \overline{\mathbb{R}}$ of a suitably structured sub-interval of the extended reals. We leave to future work the more general treatment where these assumptions are relaxed, e.g., A is compact metrizable is a standard assumption for $\mathbf{C}_0 A$.

Our development differs from the classical theory:

Classical theory	Here
• Equip the space with a topology or a σ-field.	• Use a suitable space, e.g., for evaluation.
• Show it is standard, typically by metrizing it.	• Exhibit an isomorphism to a standard space.
• Show compatibility with operations of interest, e.g. evaluation is measurable.	• Optionally, study a compatible metric.

While in both cases begin by equipping the points of interest with a space structure, in our case we have more constructions available, and so we choose a space that supports the operations of interest from the outset. The remaining properties and structure—concreteness and a compatible metric or topology—can be deferred to a later point, if and when they are needed. Moreover, type-driven measurability may further simplify how we derive these structures and properties.

We define the space of continuous functions $\mathbf{C}_0 I \hookrightarrow \overline{\mathbb{R}}^I$ as a subspace of the function space. By fiat, it supports a measurable evaluation operation. In the remainder of this subsection we will show that $\mathbf{C}_0 I$ is concrete, and that it supports the classical compatible metric.

Theorem 6. *Let $I \subseteq \overline{\mathbb{R}}$ be a non-degenerate interval. The space of continuous functions $\mathbf{C}_0 I \hookrightarrow \overline{\mathbb{R}}^I$ is standard, and it embeds into the standard space of rationally-indexed sequences via restriction:*

$$(-|_{I\cap\mathbb{Q}}) := \lambda f.\lambda q.f\,q : \mathbf{C}_0 I \hookrightarrow \overline{\mathbb{R}}^{I\cap\mathbb{Q}}$$

The crux of the proof involves a Boolean predicate $\varphi : \overline{\mathbb{R}}^{I\cap\mathbb{Q}} \to \mathbb{B}$ that characterises those sequences that can be uniquely extended to a continuous funcion. The core is this property:

Theorem (Heine-Cantor)**.** *Consider a function $f : \llcorner A \lrcorner \to \llcorner B \lrcorner$ between the points of two metric spaces. If A is compact, then f is continuous iff it is uniformly continuous.*

While $I \subseteq [-\infty, \infty]$ may not be compact, i.e., closed, we can use the Heine-Cantor theorem on all the rationally-indexed closed sub-intervals $[p, q] \subseteq I$. Formally, let $a := \inf I, b := \sup I$ and:

$$\varphi\vec{y} := \forall p, q \in \mathbb{Q} \cup \{a, b\} .p, q \in I \implies \forall \varepsilon \in \mathbb{Q} \cap (0, \infty).\exists\delta \in \mathbb{Q} \cap (0, \infty).\forall x_1, x_2 \in \mathbb{Q} \cap I.$$
$$x_1, x_2 \in [p, q] \wedge |x_1 - x_2| \leq \delta \implies |fx_1 - fx_2| \leq \varepsilon$$

By type-checking, this predicate is measurable, and by the Heine-Cantor theorem, if $f : I \to \overline{\mathbb{R}}$ is continuous, then $\varphi(f|_{I\cap\mathbb{Q}})$. For the inverse function, let $\vec{q} : \mathbb{N} \twoheadrightarrow \mathbb{Q}$ be any enumeration:

$$\psi : \left\{\vec{y} \in \overline{\mathbb{R}}^{I\cap\mathbb{Q}} \middle| \varphi(\vec{y})\right\} \to \overline{\mathbb{R}}^{I} \qquad \psi\vec{y} := \lambda x.\,\mathbf{let}\ \vec{k} : (I \cap \mathbb{Q})^{\mathbb{N}} = \lambda n.\mathcal{Z}(\vec{q}, I \cap (x - \tfrac{1}{n}, x + \tfrac{1}{n}))\ \mathbf{in}\ \lim_{n} y_{k_n}$$

We select, for each $x \in I$, a sequence of rationals converging to x and take the limit. By type-checking, this function is measurable. The predicate φ ensures each function $\varphi\vec{y}$ is also continuous. Indeed, for every $x \in I$, there is a closed neighbourhood $x \in [p, q] \subseteq I$ with $p, q \in \mathbb{Q} \cup \{a, b\}$. By $\varphi\vec{y}$, and moving to the limit, we have that $\psi\vec{y}$ is uniformly continuous on $[p, q]$, and so continuous at x. Thus ψ lifts to a function into $\mathbf{C}_0 I$, and a standard argument shows it is the inverse to $(-|_{I\cap\mathbb{Q}})$.

We still need to recover one more main result from the classical theory, namely characterising the uniform metric topology as generating the σ-field when I is compact, i.e., closed.

Theorem 7. *For $I = [a, b] \subseteq \mathbb{R}$, the* uniform *metric is compatible, and has measurable limits:*

$$d : \mathbf{C}_0 I \to \overline{\mathbb{R}} \qquad d(f, g) := \sup_{x\in I} |fx - gx|$$

It generates the σ-field of the standard subspace of finitely bounded *continuous functions.*

We have thus recovered the main results concerning $\mathbf{C}_0 I$ and $\mathbf{BC}_0 I$: they have measurable evaluation function, they are both standard, and $\mathbf{BC}_0 I$ is metrizable by the uniform metric. Our position is that this order is the order of importance for modelling: we first want to define the space and its core operations, then establish it is concrete, and optionally characterise all the measurable events. Even if we cannot establish the later stages, characterising the events or even establishing it is concrete, we can still use the space and its associated operations for modelling.

7 THE SKOROKHOD SPACE OF CÀDLÀG FUNCTIONS

The right-continuous with left-limits functions (rcll) are a relatively wide class of functions that arise in various modelling scenarios. We adopt the French terminology, continue à droite, limite à gauche (càdlàg). For example, the cumulative probability functions of a probability measure over $\overline{\mathbb{R}}$ is a well-behaved càdlàg function. As another example, the càdlàg functions take a central place in continuous-time continuous-space process theory. For example, Kallenberg [2001, p. 134] defines the class of Lévy processes, which includes Brownian motion, as processes whose outcomes are càdlàg functions. Here we will use our techniques to recover the well-known fact that càdlàg functions form a concrete space, the Skorokhod space, and a separable metric generating its events.

7.1 Definition and basic properties

Let $I := [a, b) \subseteq \mathbb{R}$ be a half-open interval. A function $f : I \to \mathbb{R}$ is *càdlàg* when, for every $t \in I$ the left limit, i.e: $f_{\to}t := \lim_{x\to t^-} fx$ exists in $\mathbb{R}$, and at every $t \in (a, b)$ the function takes the right limit: $ft = f_{\leftarrow}t := \lim_{t^+\leftarrow x} fx$. Let $\mathbf{D}I \hookrightarrow \mathbb{R}^I$ be the space of càdlàg functions over I.

Example 79 (indicator functions, continuous functions, positive-affine combinations)**.** An indicator function for an interval $J \subseteq I$ is càdlàg iff $J = [x, y)$ is half-open. Every continuous function is càdlàg. The càdlàg functions are closed under affine combinations with positive coefficients, i.e., if

f_i are càdlàg and $w_i \in \mathbb{W}$ are non-negative, then $\lambda x.a + \sum_{i=1}^{n} w_i f_i x$ is càdlàg. More generally, if $\vec{f}$ are càdlàg and $g : \mathbb{R}^n \to \mathbb{R}$ is monotone and continuous, then $g \circ \vec{f}$ is càdlàg. □

Example 80. A *cumulative probability function* is a monotone càdlàg function $F : I \to \mathbb{W}$ that vanishes at a, i.e. $Fa = 0$ and is moreover monotone. We will not study these functions further. □

Evaluation and taking left limits are measurable functions eval, $\lambda(f, x).f_{\to}x : \mathbf{D}I \times I \to \overline{\mathbb{R}}$.

Let $f : I \to \mathbb{R}$ be a function, and $\varepsilon > 0$ a real number. An *ε-discontinuity* of f is a point $t \in I$ for which $|f_{\to}t - f_{\leftarrow}t| \geq 0$. The *interval partitions* of I are the lists in the following subspace:

$$\mathbf{IPart}_I \coloneqq \left\{\vec{t} \in \mathbf{List}\,\mathbb{R} \middle| \mathbf{let}\,\vec{s} := a :- \vec{t} -: b \ \mathbf{in}\ \forall i.0 \leq i \leq \mathbf{length}\,\vec{t} \Rightarrow s_i < s_{i+1}\right\} \hookrightarrow \mathbf{List}\,\mathbb{R}$$

We think of each list in $[t_1, \ldots, t_n] \in \mathbf{IPart}_I$ as the partition $a =: t_0 < t_1 \ldots, t_n < t_{n+1} := b$, and call each interval $[t_i, t_{i+1})$, for $i = 0, \ldots, n$ a *segment* of the partition $\vec{t}$, and each point t_i, $1 \leq i \leq n$ an *intermediate* point in the partition. We say that $f : I \to \mathbb{R}$ is *ε-tight* in a partition $\vec{t}$ when, for every segment we have $\forall x, y \in [t_i, t_{i+1}).\, |fx - fy| \leq \varepsilon$. When $\vec{t}$ is empty, it is *immediately* ε-tight.

Example 81. The ε-discontinuities, $0 < \varepsilon \leq 1$, for each characteristic function $[- \in [x, y)]$ are x and y, and the function is ε-tight in the length 2 partition $[x, y]$ whose segments are $[a, x)$, $[x, y)$, and $[y, b)$. This function has no ε-discontinuities and it is immediate ε-tight in I for $\varepsilon > 1$. □

If f is ε-tight in $\vec{t}$, then $\vec{t}$ must contain every ε-discontinuity as an intermediate point. Therefore, an ε-tight function has finitely many ε-discontinuities. The following characterisation for the càdlàg property, a modification of a result by Davidson [2021, Thm. 3.20], is the crux for the concreteness of the Skorokhod space:

Proposition 8. *For $f : I \to \mathbb{R}$ the following conditions are equivalent: f is càdlàg; f is ε-tight for all $\varepsilon > 0$; there is an interval partition $\vec{t}$ in which f is ε-tight and every intermediate point is either rational or an ε-discontinuity for all $\varepsilon > 0$.*

7.2 Measurability toolkit

To exhibit the Skorokhod space as concrete, we first develop some tools. We have Boolean predicates for a càdlàg function having an ε-discontinuity at a given point and being immediately ε-tight. Let $\mathbf{Interval} \coloneqq \left\{(x, y) \in [a, b]^2 \middle| x < y\right\}$:

$$\text{-discontinuousAt} : (0, \infty) \times \mathbf{D}I \times (a, b) \to \mathbb{B} \qquad \varepsilon\text{-discontinuousAt}(f, t) \coloneqq |f_{\to}t - ft| \geq \varepsilon$$
$$\text{-immTight} : (0, \infty) \times \mathbf{D}I \times \mathbf{Interval} \to \mathbb{B}$$
$$\varepsilon\text{-immTight}(f, x, y) \coloneqq \forall p, q \in \mathbb{Q}.p, q \in [x, y) \Rightarrow |fp - fq| \leq \varepsilon$$

Both predicates rely on f already being càdlàg. We use them to check whether a given càdlàg function is ε-tight in a given partition:

$$\text{-tight} : (0, \infty) \times \mathbf{D}I \times \mathbf{IPart}_I \to \mathbb{B}$$
$$\varepsilon\text{-tight}(f, \vec{t}) \coloneqq \mathbf{let}\,\{\vec{s} = a :- \vec{t} -: b; n = \mathbf{length}\,\vec{t}\}\ \mathbf{in}\ \forall i.0 \leq i \leq n \Rightarrow \varepsilon\text{-immTight}(f, (s_i, s_{i+1}))$$

We define the following Boolean predicate ε-discontinuity$(f, (x, y))$ that characterises whether f has an ε-discontinuity in $[x, y)$:

$$\text{-discontinuity} : (0, \infty) \times \mathbf{D}I \times \mathbf{Interval} \to \mathbb{B}$$
$$\varepsilon\text{-discontinuity}(f, (x, y)) \coloneqq \nexists \vec{t} \in \mathbf{List}\,\mathbb{Q}.\vec{t} \in \mathbf{IPart}_{(x,y)} \wedge \varepsilon\text{-tight}(f, \vec{t}) \wedge$$
$$\forall i.1 \leq i \leq \mathbf{length}\,\vec{t} \Rightarrow \neg\varepsilon\text{-discontinuousAt}(f, t_i)$$

Indeed, if there are no ε-discontinuities, then by Proposition 8 there is an interval partition $\vec{t}$ consisting of rational internal points and no ε-discontinuities, so witnesses that the predicate fails.

Conversely, if the predicate fails, there is a witnessing partition $\vec{t}$. If there was an ε-discontinuity, then it must appear in $\vec{t}$, contradicting $\vec{t}$'s defining property.

We can therefore select the left-most ε-discontinuity in a given sufffix, if it exists:

$$\text{Ƨ} : \{(\varepsilon, f, x) \in (0, \infty) \times \mathbf{D}I \times I\} \rightharpoonup \left\{ s \in I \middle| \begin{array}{l} \varepsilon\text{-discontinuousAt}(f, s), \\ \forall t \in [a, s).\neg\varepsilon\text{-discontinuousAt}(f, t) \end{array} \right\}$$

$$\text{Ƨ}(\varepsilon, f, x) \coloneqq \mathbf{if}\ \varepsilon\text{-discontinuity}(f, (x, b))\ \mathbf{then}\ \inf\left(a < q < b \wedge \varepsilon\text{-discontinuity}(f, (x, q))\right)_{q \in \mathbb{Q}}\ \mathbf{else}\ \bot$$

Indeed, by fiat the domain of definition is those suffixes in which there is an ε-discontinuity. For those, there are only finitely many ε-discontinuities since f is càdlàg, and the first one is the infimum of those rationals that have an ε-discontinuity to their left.

We can use this selection principle to enumerate all the ε-discontinuities. We use the coinductive structure of the space of finite-or-infinite sequences $\mathbf{List}\,\mathbb{R} \amalg \mathbf{Stream}\,\mathbb{R}$:

$$\mathbf{List}\,\mathbb{R} \amalg \mathbf{Stream}\,\mathbb{R} \cong \boldsymbol{\nu}\mathbf{CStream}\,\mathbb{R}.\{\mathbf{Nil}, \mathsf{Cons} : (\mathbb{R}, \mathbf{CStream}\,\mathbb{R})\}$$

We define the enumeration function coinductively:

$$\text{-discontinuities} : \{(\varepsilon, f) \in (0, \infty) \times \mathbf{D}I\} \to \left\{\vec{t} \in \mathbf{IPart}_I \middle| \vec{t} \text{ enumerates } f\text{'s } \varepsilon\text{-discontinuities}\right\}$$

$$\varepsilon\text{-discontinuities} f \coloneqq$$

$$\mathbf{case}\ \mathbf{CStream}\,\mathbb{R}.\mathbf{unfold} \begin{pmatrix} \lambda(\varepsilon, f, x).\mathbf{if}\ \varepsilon\text{-discontinuity}(f, (x, b)) \\ \qquad \mathbf{then}\ \mathbf{let}\ t = \text{Ƨ}(\varepsilon, f, x)\ \mathbf{in}\ \mathsf{Cons}(t, (\varepsilon, f, t)) \\ \qquad \mathbf{else}\ \mathbf{Nil} \end{pmatrix} (\varepsilon, f, a)\ \mathbf{of} \begin{cases} \iota_1 \vec{t}.\vec{t}; \\ \iota_2 \_.\bot \end{cases}$$

The seed (ε, f, x) propagate the given ε and function, alongside a point $x \in I$ after which we want to find an ε-discontinuity in (x, b). Starting with $x = a$, we check whether there is another ε-discontinuity. If there is not, we terminate the coinduction with the empty list **Nil**. Otherwise, we select the left-most ε-discontinuity t, and resume enumerating all the discontinuities in (t, b). A-priori this enumeration may continue indefinitely and result in an infinite stream, and so this function may seem partial. Since a càdlàg function only has finitely many ε-discontinuities, we will always terminate with a list, and so this function is total.

7.3 Concreteness

To exhibit the Skorokhod space as standard, we mimic Theorem 6 and represent each function by its restriction to values at rational points. We recover the original function by taking right-limits along any rational sequence. We need additional data in order to characterise those sequences that extend to a càdlàg functions. We want to validate that the extended function will be ε-tight for some partition for all ε. Any such partition will include all the ε-discontinuities, and these may occur at any point, not only on rational ones, and so we are not guaranteed to be able to merely select them from the rational sequence. Instead, we require the ε-discontinuities as data.

Theorem 9. *The Skorokhod space is standard, represented by values at rationals and ε-discontinuities:*

$$\lambda f.\left(f\Big|_{I\cap\mathbb{Q}}, (\varepsilon\text{-discontinuities} f)_{\varepsilon} \right) : \mathbf{D}I \hookrightarrow \mathbb{R}^{I\cap\mathbb{Q}} \times \mathbf{IPart}_I^{(0,\infty)\cap\mathbb{Q}}$$

7.4 Compatible metrisation

The uniform metric is compatible with the Skorokhod space, but it is not separable over it. Kolmogorov [1956] proposed a different metric, which is separable and, as we will show, compatible with the Skorokhod space and has measurable limits, and therefore generates its σ-field.

The uniform metric is compatible since, for two càdlàg functions $f, g \in \mathbf{D}I$, we have:

$$\sup_{x\in I} |fx - gx| = \sup_{q \in I\cap\mathbb{Q}} |fq - gq|$$

Example 82 (Jacod and Shiriyaev 2003, p. 325)**.** The uniform metric $d_{\mathbf{U}}$ is not separable. Consider the family of càdlàg functions $\vec{f} : (0,1) \to \mathbf{D}[0,1)$, $f_t := [- \in [0,t)]$. They are at uniform distance 1 from each other: $t \neq s \implies d_{\mathbf{U}}(f_t, f_s) = 1$. Take any countable sequence $\vec{g}$ as a putative dense subset. For each n, either for all t, we have $d_{\mathbf{U}}(f_t, g_n) \geq \frac{1}{2}$, or there is some t_n with $d_{\mathbf{U}}(f_{t_n}, g_n) < \frac{1}{2}$. By cardinality, there is some $s \in (0,1) \setminus \{t_n | n \in \mathbb{N}\}$, so $d_{\mathbf{U}}(f_s, g_n) > \frac{1}{2}$. □

Instead, Kolmogorov [1956] proposed the following metrization for Skorokhod's topology [1956]. We say that f and g are *ε-Skorokhod-similar*, $f \overset{\varepsilon}{\sim} g$, when there are two interval partitions $\vec{t}^f, \vec{t}^g \in \mathbf{IPart}_I$ of the same length $n := \mathbf{length}\,\vec{t}^f = \mathbf{length}\,\vec{t}^g$, satisfying:

$$\forall 0 \leq i \leq \mathbf{length}\,\vec{t}^f.\, \left|t_i^f - t_i^g\right| \leq \varepsilon \wedge \forall x \in [t_i^f, t_{i+1}^f), y \in [t_i^g, t_{i+1}^g).\, |fx - gy| \leq \varepsilon$$

We then define the *Skorokhod metric* by $d_{\mathbf{S}}(f,g) = \inf\left\{\varepsilon \in [0,\infty] \middle| f \overset{\varepsilon}{\sim} g\right\}$.

Aside. The Skorokhod topology is often metrized using a metric informally described as 'continuous wiggle in time', which we otherwise will not use here. Let Λ be the set of monotone continuous homeomorphisms $\varphi : I \xrightarrow{\cong} I$. This metric metrizes the Skorokhod topology:

$$d'_{\mathbf{S}}(f,g) = \inf_{\varphi \in \Lambda} \max(d_{\mathbf{U}}(f, \mathrm{id}), d_{\mathbf{U}}(f, g \circ \varphi))$$

Proposition 10. *The Skorokhod metric $d_{\mathbf{S}}$ is compatible with* $\mathbf{D}I$*, and has measurable limits. Since it is separable, it generates the events in the Skorokhod space: $\mathcal{B}_{d_{\mathbf{S}}} = \mathcal{B}_{\mathbf{D}I}$.*

8 CONCLUSION

We have applied the theory of quasi-Borel spaces to establish that concrete spaces of functions, the continuous-function space and Skorokhod space, embed into the space of measurable functions. The methodology, use types formers to construct a space supporting the operations of interest by construction, allows us to put together simple measurability building blocks—limits, sequences, selection principles—to construct fairly complex measurability proofs.

While the development is rife with connections to classical measure theory, we took care to present the results in a self-contained manner, without relying on classical measure theory. The area benefiting from the classical theory the most is, unsurprisingly, the treatment of events. Even there, it was useful to be able to validate that an event is measurable 'at a glance'—by type-checking its comprehension predicate is a well-typed Boolean predicate.

We have chosen not to treat probability here—measures, integrals, expectation, independence and conditioning. These benefit from typed measurability proofs, especially the treatment of random variable spaces and their modes of convergence, conditional expectation, martingale theory, etc. The additional technical development introduces more cognitive overhead. Moreover, the treatment of probability would benefit from more sophisticated types—quotients and even dependent types, and from a synthetic treatment, which deserves its own focus. Since the paper introducing quasi-Borel spaces [Staton et al. 2017] and its follow-up literature [Dash et al. 2023; Ścibior et al. 2018; Vákár et al. 2019; Vandenbroucke and Schrijvers 2019] accounts for probability, we chose to let probability take a back seat and give the spotlight to measurability.

Similarly, we have not treated spaces of syntax. Example 62 (probabilistic context-free grammars) suggestively connects datatypes with measurable representations of abstract syntax. However, realistic syntax also involves binding and substitution. We believe the theory of quasi-Borel spaces is well-suited for semantic theories in the style of Fiore, Plotkin, and Turi [Fiore 2008; Fiore et al. 1999, e.g.]. We leave such a development to the future.

ACKNOWLEDGMENTS

This work was funded by *anonymized for peer review*. For the purpose of Open Access the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

REFERENCES

Robert J. Aumann. 1961. "Borel structures for function spaces." *Illinois Journal of Mathematics*, 5, 614–630.

Giorgio Bacci, Robert Furber, Dexter Kozen, Radu Mardare, Prakash Panangaden, and Dana S. Scott. 2018. "Boolean-Valued Semantics for the Stochastic λ-Calculus." In: *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, Oxford, UK, July 09-12, 2018*. Ed. by Anuj Dawar and Erich Grädel. ACM, 669–678. DOI: 10.1145/3209108.3209175.

Fredrik Dahlqvist and Dexter Kozen. 2020. "Semantics of higher-order probabilistic programs with conditioning." *Proc. ACM Program. Lang.*, 4, POPL, 57:1–57:29. DOI: 10.1145/3371125.

Swaraj Dash, Younesse Kaddar, Hugo Paquet, and Sam Staton. Jan. 2023. "Affine Monads and Lazy Structures for Bayesian Programming." *Proc. ACM Program. Lang.*, 7, POPL, Article 46, (Jan. 2023), 31 pages. DOI: 10.1145/3571239.

James Davidson. Nov. 2021. "668Càdlàg Functions." In: *Stochastic Limit Theory: An Introduction for Econometricians*. Oxford University Press, (Nov. 2021). ISBN: 9780192844507. eprint: https://academic.oup.com/book/0/chapter/355983890/chapter-pdf/46987921/oso-9780192844507-chapter-30.pdf. DOI: 10.1093/oso/9780192844507.003.0030.

Thomas Ehrhard, Michele Pagani, and Christine Tasson. 2018. "Measurable cones and stable, measurable functions: a model for probabilistic higher-order programming." *Proc. ACM Program. Lang.*, 2, POPL, 59:1–59:28. DOI: 10.1145/3158147.

Marcelo Fiore. 2008. "Second-Order and Dependently-Sorted Abstract Syntax." In: *2008 23rd Annual IEEE Symposium on Logic in Computer Science*, 57–68. DOI: 10.1109/LICS.2008.38.

Marcelo Fiore, Gordon Plotkin, and Daniele. Turi. 1999. "Abstract syntax and variable binding." In: *Proceedings. 14th Symposium on Logic in Computer Science (Cat. No. PR00158)*, 193–202. DOI: 10.1109/LICS.1999.782615.

J. A. Goguen, J. W. Thatcher, E. G. Wagner, and J. B. Wright. Jan. 1977. "Initial Algebra Semantics and Continuous Algebras." *J. ACM*, 24, 1, (Jan. 1977), 68–95. DOI: 10.1145/321992.321997.

Jean Goubault-Larrecq, Xiaodong Jia, and Clément Théron. 2023. "A Domain-theoretic Approach to Statistical Programming Languages." *J. ACM*, 70, 5, 35:1–35:63. DOI: 10.1145/3611660.

Chris Heunen, Ohad Kammar, Sam Staton, and Hongseok Yang. 2017. "A convenient category for higher-order probability theory." In: *LiCS*. http://ieeexplore.ieee.org/document/8005137/.

Jean. Jacod and A. N. (Alʹbert Nikolaevich) Shiriyaev. 2003. *Limit theorems for stochastic processes / Jean Jacod, Albert N. Shiryaev*. eng. (Second edition. ed.). Grundlehren der mathematischen Wissenschaften ; 288. Springer-Verlag, Berlin ; ISBN: 3540439323.

David Janz, Brooks Paige, Tom Rainforth, Jan-Willem van de Meent, and Frank Wood. 2016. *Probabilistic structure discovery in time series data*. (2016). arXiv: 1611.06863 [stat.ML].

Xiaodong Jia, Bert Lindenhovius, Michael W. Mislove, and Vladimir Zamdzhiev. 2021. "Commutative Monads for Probabilistic Programming Languages." In: *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*. IEEE, 1–14. DOI: 10.1109/LICS52264.2021.9470611.

Olav Kallenberg. 2001. *Foundations of modern probability*. Ed. by J. Gani, C.C. Heyde, and T.G. Kurtz. (2nd ed.). Probability and its applications. Vol. 77. Springer.

A.S. Kechris. 1995. *Classical Descriptive Set Theory*. Graduate Texts in Mathematics. Springer-Verlag. ISBN: 9780387943749.

A. N. Kolmogorov. 1956. "On Skorokhod Convergence." *Theory of Probability & Its Applications*, 1, 2, 215–222. eprint: https://doi.org/10.1137/1101017. DOI: 10.1137/1101017.

Marcin Sabok, Sam Staton, Dario Stein, and Michael Wolman. 2021. "Probabilistic programming semantics for name generation." *Proc. ACM Program. Lang.*, 5, POPL, 1–29. DOI: 10.1145/3434292.

Adam Ścibior et al.. 2018. "Denotational Validation of Higher-Order Bayesian Inference." *Proceedings of the ACM on Programming Languages*, POPL.

A. V. Skorokhod. 1956. "Limit Theorems for Stochastic Processes." eng. *Theory of probability and its applications*, 1, 3, 261–290.

Matthijs Vákár, Ohad Kammar, and Sam Staton. Jan. 2019. "A domain theory for statistical probabilistic programming." *Proc. ACM Program. Lang.*, 3, POPL, Article 36, (Jan. 2019), 29 pages. DOI: 10.1145/3290349.

Alexander Vandenbroucke and Tom Schrijvers. Dec. 2019. "Functional probabilistic NetKAT." *Proc. ACM Program. Lang.*, 4, POPL, Article 39, (Dec. 2019), 27 pages. DOI: 10.1145/3371107.

$$
\begin{array}{llll}
\textbf{Type} & \ni A, B := & & \text{types} \\
 & & A \to B & \quad \text{function type} \\
 & | & \dashv\!|X|\!\vdash & \quad \text{semantic type reflection} \\
 & | & \{\rho\} & \quad \text{variant type } (\mathrm{Dom}\,\rho \text{ countable}) \\
 & | & \lhd\rho\rhd & \quad \text{record type } (\mathrm{Dom}\,\rho \text{ finite}) \\
\textbf{preTerm} & \ni M, N := & & \qquad\qquad\qquad\qquad \text{terms} \\
 & & x & \quad \text{variable} \\
 & | & \lambda x : A.M & \quad \text{abstraction} \\
 & | & M\,N & \quad \text{application} \\
 & | & \textbf{let}\, x = M \;\textbf{ in }\; N & \quad \text{intermediate result} \\
 & | & \dashv\!|f : G \to X|\!\vdash_A & \quad \text{semantic reflection} \\
 & | & A.\ell\, M & \quad \text{data constructor} \\
 & | & \textbf{case}\, M\, \textbf{of}\, \{\ell_\ell x_\ell.M_\ell | \ell \in I\} & \quad \text{pattern match} \\
 & | & \boldsymbol{<}\ell_1 : M_1, \ldots, \ell_n : M_n\boldsymbol{>} & \quad \text{record} \\
 & | & \textbf{case}\, M\, \textbf{of}\, \boldsymbol{<}\ell : x_\ell | \ell \in I\boldsymbol{>}.N & \quad \text{pattern match}
\end{array}
$$

Fig. 7. Full simply-typed language

$$
\frac{(x : A) \in \Gamma}{\Gamma \vdash x : A}
\qquad
\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x : A.M : B}
\qquad
\frac{\Gamma \vdash M : A \to B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\,N : B}
$$

$$
\frac{\Gamma \vdash M : A \qquad \Gamma, x : A \vdash N : B}{\Gamma \vdash \textbf{let}\, x = M \;\textbf{ in }\; N : B}
\qquad
\frac{\Gamma \text{ denotes } G \qquad A \text{ denotes } X \qquad f : G \to X \text{ measurable}}{\Gamma \vdash \dashv\!|f : G \to X|\!\vdash_A : A}
$$

$$
\frac{(\ell : A) \in \rho \qquad \Gamma \vdash M : A}{\Gamma \vdash \{\rho\}.\ell\, M : \{\rho\}}
\qquad
\frac{\Gamma \vdash M : \{\rho\} \qquad \text{for all } (\ell : A) \in \rho\text{: } \Gamma, x_\ell : A_\ell \vdash M :}{\textbf{case}\, M\, \textbf{of}\, \{\ell_\ell x_\ell.M_\ell | \ell \in \mathrm{Dom}\rho\}}
$$

$$
\frac{\text{for all } \ell \in I\text{: } \Gamma \vdash M_\ell : A_\ell}{\Gamma \vdash \boldsymbol{<}\ell := M_\ell | \ell \in I\boldsymbol{>} : \lhd\ell : A_\ell | \ell \in I\rhd}
\qquad
\frac{\Gamma \vdash M : \lhd\ell : A_\ell | \ell \in I\rhd \qquad \Gamma, (x_\ell : A_\ell)_{\ell \in I} \vdash N}{\Gamma \vdash \textbf{case}\, M\, \textbf{of}\, \boldsymbol{<}\ell : x_\ell | \ell \in I\boldsymbol{>}.N : B}
$$

Fig. 8. Simple-type system: all rules

A SIMPLY TYPED LANGUAGE IN FULL

For convenience, we collect all the constructs of the simple language in one place. Fig. 7 presents the full syntax for the simply-typed language. Fig. 8 presents the typing rules. Fig. 9 presents the denotations of simple types and contexts, and Fig. 10 presents the denotations of well-formed terms. We unfortuantely were not able to include the treatment of mixed inductive types in time for submission, but we hope to finish type-setting them for future versions.

B DETAILED MEASURABILITY PROOFS

For completeness, we include fairly detailed proofs for the various analytical results from §6–§7.

B.1 The continuous-function space

We include the proof sketch for the following theorem:

$$\llbracket A \to B \rrbracket := \llbracket B \rrbracket^{\llbracket A \rrbracket} \qquad \llbracket \dashv\!|X \Vdash \rrbracket := X \qquad \llbracket \{\rho\} \rrbracket := \coprod_{(\ell:A)\in\rho} \llbracket A \rrbracket \qquad \llbracket \lhd\rho\rhd \rrbracket := \prod_{(\ell:A)\in\rho} \llbracket A \rrbracket$$

$$\llbracket \Gamma \rrbracket := \prod_{(x:A)\in\Gamma} \llbracket A \rrbracket$$

Fig. 9. Denotations of simple types

$$\llbracket x \rrbracket \gamma := \pi_x \gamma \qquad \llbracket \lambda x : A.M \rrbracket \gamma := \lambda u. \llbracket M \rrbracket (\gamma, u) \qquad \llbracket M\, N \rrbracket \gamma := \mathsf{eval}\,(\llbracket M \rrbracket \gamma, \llbracket N \rrbracket \gamma)$$

$$\llbracket \mathbf{let}\, x = M \;\mathbf{in}\; N \rrbracket \gamma := \llbracket N \rrbracket (\gamma, \llbracket M \rrbracket \gamma) \qquad \llbracket \dashv\!|f : \llbracket \Gamma \rrbracket \to \llbracket A \rrbracket \Vdash_A \rrbracket \gamma := f\,\gamma$$

$$\llbracket A.\ell\, M \rrbracket \gamma := \iota_\ell \llbracket M \gamma \rrbracket \qquad \llbracket \mathbf{case}\, M\, \mathbf{of}\, \{\ell_\ell x_\ell.M_\ell | \ell \in I\} \rrbracket \gamma := [\llbracket M_\ell \rrbracket]_{\ell \in I} (\gamma, \llbracket M \rrbracket \gamma)$$

$$\llbracket \mathbf{<}\ell_1 : M_{\ell_1}, \ldots, \ell_n : M_{\ell_n} \mathbf{>} \rrbracket \gamma := (\llbracket M_\ell \rrbracket \gamma)_{\ell \in \{\ell_1,\ldots,\ell_n\}} \quad \llbracket \mathbf{case}\, M\, \mathbf{of}\, \mathbf{<}\ell : x_\ell | \ell \in I\mathbf{>}.N \rrbracket \gamma := \llbracket N \rrbracket (\gamma, \llbracket M \rrbracket \gamma)$$

Fig. 10. Denotations of simply-typed terms

Theorem 7. *For $I = [a, b] \subseteq \mathbb{R}$, the* uniform *metric is compatible, and has measurable limits:*

$$d : \mathbf{C}_0 I \to \overline{\mathbb{R}} \qquad d(f, g) := \sup_{x \in I} |fx - gx|$$

It generates the σ-field of the standard subspace of finitely bounded *continuous functions.*

Proof

For compatibility, note $d(f, g) = \sup_{q \in I \cap \mathbb{Q}} |fq - gq|$ and so the metric is compatible. Limits for d are uniformly-convergent sequences, which also converge pointwise, and so $\lim_n f_n = \lambda x. \lim_n f_n x$ is measurable over its domain of definition. The uniform metric is Cauchy-complete, and so the uniform metric has measurable limits.

The subspace of finitely bounded continuous functions is therefore Borel:

$$\mathbf{BC}_0 I := \left\{ f \in \mathbf{C}_0 I \,\middle|\, \sup_{x \in I} |fx| < \infty \right\} \hookrightarrow \mathbf{C}_0 I$$

Since the uniform metric is separable on $\mathbf{BC}_0 I$, e.g., by Weierstrass's approximation theorem, we conclude by Proposition 5. ∎

B.2 The Skorokhod space: concreteness

The first proof is for the following characterisation of càdlàgfunctions.

Proposition 8. *For $f : I \to \mathbb{R}$ the following conditions are equivalent: f is càdlàg; f is ε-tight for all $\varepsilon > 0$; there is an interval partition $\vec{t}$ in which f is ε-tight and every intermediate point is either rational or an ε-discontinuity for all $\varepsilon > 0$.*

Proof

This result is a modification of a result by Davidson [2021, Thm. 3.20]. The third condition strengthens the second, so implies it. Assume f is ε-tight for every ε. Take any $s \in I$. For every ε, there is some segment with $s \in [t_i^\varepsilon, t_{i+1}^\varepsilon)$ in which f is immediately ε-tight. The right-neighborhoods $[s, t_{i+1}^\varepsilon)$ show that $f_{\leftarrow} s = fs$. Similarly, take any $s \in (a, b)$. For every ε, there is some segment with

$$\varphi, \psi : \mathbb{R}^{I \cap \mathbb{Q}} \times (0, \infty) \times \mathbf{IPart}_I \to \mathbb{B} \qquad \mathbf{D}_{\mathrm{code}} I \coloneqq \left\{ (\vec{y}, \vec{\vec{t}}) \middle| \forall \varepsilon \in (0, \infty) \cap \mathbb{Q}. \varphi(\vec{y}, \varepsilon, \vec{t}_\varepsilon) \land \psi(\vec{y}, \varepsilon, \vec{t}_\varepsilon) \right\}$$

$$\varphi(\vec{y}, \varepsilon, \vec{t}) \coloneqq \mathbf{let}\, \vec{s} = a :\!- \vec{t} -\!: b \;\; \mathbf{in}\; \forall i. 0 \le i \le \mathbf{length}\, \vec{t} \Rightarrow \varepsilon\text{-rationallyTight}(f, (s_i, s_{i+1}))$$

$$\psi(\vec{y}, \varepsilon, \vec{t}) \coloneqq \mathbf{let}\, \vec{s} = a :\!- \vec{t} -\!: b \;\; \mathbf{in}\; \forall i. 1 \le i \le \mathbf{length}\, \vec{t}.$$

$$\mathbf{let}\, \{\vec{q}^{\,-} = \text{Ƨ}_{\to}(s_{i-1}, s_i); \vec{q}^{\,+} = \text{Ƨ}_{\leftarrow}(s_i, s_{i+1})\} \;\; \mathbf{in}\; \lim_n \left| y_{q_n^+} - y_{q_n^-} \right| \ge \varepsilon$$

Fig. 11. A measurable characterisation for the image of the embedding

$(t_i^\varepsilon, s) \subseteq (t_i^\varepsilon, t_{i+1}^\varepsilon)$. Selecting a rational q_n from each $(t_i^{\frac{1}{n}}, s)$ yields: (a) a Cauchy sequence $f q_n$, say with limit L; a punctured left-neighborhood for each $\varepsilon \coloneqq \frac{1}{n}$ exhibiting $f_{\to} s = L$. Thus f is càdlàg.

Next, assume f is càdlàg, and consider any $\varepsilon > 0$. Let S be the set of reals $r \in [a, b]$ for which there exists an interval partition of $[a, r)$ that contains only ε-discontinuities or rationals as internal points and in which f is ε-tight. This set is not empty: since $f a = f_{\leftarrow} a$, there is some right-neighborhood $[a, r)$ of a in which f is immediately ε-tight. The empty list witnesses $r \in S$.

Let $\tau \coloneqq \sup S \le b$. Then $\tau > a$, since S contains some $r > a$. Moreover, $\tau \in S$. Indeed, since $f_{\to} \tau$ exists, there is some punctured left-neighborhood (s, τ) in which f is immediately ε-tight. Since $s < \tau = \sup S$, there is some $s < r \in S$ with a witnessing partition $\vec{t}$, $n \coloneqq \mathbf{length}\, \vec{t}$. The partition $\vec{t} -\!: r$ is nearly appropriate to exhibit $\tau \in S$. The function is tight in it, but r might be neither rational nor an ε-discontinuity. Pick a rational $q \in (t_n, r)$. We then have that $\vec{t} -\!: q$ is a partition in which every internal point is rational or an ε-discontinuity. Since and $[t_n, q) \subseteq [t_n, r)$ and $[q, \tau) \subseteq (s, \tau)$, we have that f is immediately tight in both of these segments. Therefore $\tau \in S$.

Finally, we have that $\tau = b$. Let $\vec{t}$, $n \coloneqq \mathbf{length}\, \vec{t}$ witness $\tau \in S$. Assume to the contrary that $\tau < b$. If τ is an ε-discontinuity, then, as $f \tau = f_{\leftarrow} \tau$, there is a right-neighborhood $[\tau, r)$ where f is immediately ε-tight, and the interval partition $\vec{t} -\!: \tau$ witnesses that $S \ni r > \tau$, a contradiction. If τ is not an ε-discontinuity, then there is: a punctured left-neighborhood (s, τ), with s rational, in which f is immediately $\frac{\varepsilon}{2}$-tight; and a right-neighborhood $[\tau, r)$ in which f is immediately $\frac{\varepsilon}{2}$-tight. We then have that f is immediately ε-tight in $[t_n, s)$ and in $[s, r)$, and so $\vec{t} -\!: s$ witnesses that $S \ni r > \tau$, a contradiction. Either way, we arrived at a contradiction, and so $\tau = b$. ∎

Next, we prove the Skorokhod space is standard:

Theorem 9. *The Skorokhod space is standard, represented by values at rationals and ε-discontinuities:*

$$\lambda f. \left(f\Big|_{I \cap \mathbb{Q}}, (\varepsilon\text{-discontinuities} f)_\varepsilon \right) : \mathbf{D} I \hookrightarrow \mathbb{R}^{I \cap \mathbb{Q}} \times \mathbf{IPart}_I^{(0,\infty) \cap \mathbb{Q}}$$

Proof

Let ρ be the embedding. We characterise its image $\mathbf{D}_{\mathrm{code}} I$ via the two predicates φ and ψ in Fig. 11. The predicate $\varphi(\vec{y}, \varepsilon, \vec{t})$ ensures $\vec{y}$ must be ε-tight in $\vec{t}$, and the predicate $\psi(f, \varepsilon, \vec{t})$ ensures $\vec{y}$ is ε-discontinuous at every t_i.

To define φ, we say that a rationally-indexed sequence $\vec{y}$ is *rationally ε-tight* at $[x, y)$ when there is an interval partition of $[x, y)$ with rational internal points for which the sequence is ε-tight, i.e.:

$$\varepsilon\text{-rationallyTight}(\vec{y}, (x, y)) \coloneqq \exists \vec{q} \in \mathbf{List}\, \mathbb{Q}. \vec{q} \in \mathbf{IPart}_{[x,y)} \Rightarrow \mathbf{let}\, \vec{p} = x :\!- x\vec{q} -\!: y\, \mathbf{in}$$
$$\forall i. 0 \le i \le \mathbf{length}\, \vec{q}. \forall r, s \in \mathbb{Q}. r, s \in [p_i, p_{i+1}) \Rightarrow |y_r - y_s| \le \varepsilon$$

To define ψ, let $\text{Ƨ}_{\to}, \text{Ƨ}_{\leftarrow} : \mathbf{Interval} \to \mathbb{Q}^{\mathbb{N}}$ be selection principles for rational sequences $\text{Ƨ}_{\to}(x, y)$, $\text{Ƨ}_{\leftarrow}(x, y)$ that converge to y from the left and to x from the right, respectively, in (x, y). The embedding lifts to a function $\xi : \mathbf{D} I \to \mathbf{D}_{\mathrm{code}} I$. Indeed, by Proposition 8, there is a partition $\vec{t}$ in which f is ε-tight consisting only of rationals and ε-discontinuities as internal points. Since ε-discontinuities f

exhaustively enumerates all the ε-discontinuities and all discontinuities must appear in $\vec{t}$, then $\vec{t}$ refines the list of ε-discontinuities. In particular, it will be rationally ε-tight between every segment in the discontinuity partition, and so φ holds. Since we enumerate only ε-discontinuities, ψ holds.

In the converse direction, define $\xi^{-1}(\vec{y}, \vec{\vec{t}}) := \lambda x.\,\mathbf{let}\ \vec{q} = \mathcal{Z}_{\leftarrow}(x, b)\ \mathbf{in}\ \lim_n y_{q_n}$. Let $f := \xi^{-1}(\vec{y}, \vec{\vec{t}})$. Take any rational ε. By Proposition 8 it suffices to find a partition for which f is ε-tight. Taking an arbitrary segment $J_i := [s_i, s_{i+1})$ of $\vec{t}_\varepsilon$, by fiat there is some rational partition $\vec{q}_i$ of J_i in which f is ε-tight. Interleaving the partitions $\vec{s}$ and $\vec{q}_i$ results in a partition in which f is ε-tight.

If f is càdlàg, we recover it from its values at rational points by taking right limits, and so $(\xi^{-1} \circ \xi)f = f$. Conversely, take any $(\vec{y}, \vec{\vec{t}}) \in \mathbf{D}_{\mathrm{code}}I$, and let $f := \xi^{-1}(\vec{y}, \vec{\vec{t}})$ and $(\vec{z}, \vec{s}) := \xi f$, and we'll show that $(\vec{z}, \vec{s}) = (\vec{y}, \vec{t})$. For each rational $q \in I$, since $\vec{y}$ is rationally ε-tight in a right-neighborhood of q for every ε, we have that $z_q = f_{\leftarrow} q = y_q$. From this fact and from ψ follows that every point in $\vec{t}_\varepsilon$ is an ε-discontinuity, and that f has no ε-discontinuities between them, and so $\vec{t}_\varepsilon$ enumerates all the ε-discontinuitities in order, i.e., $\vec{s} = \vec{t}$. ■

B.3 The Skorokhod metric

We characterise strict bounds on Komogorov's Skorokhod-similarity relation in a measurable way:

Lemma 11. *Let $f, g \in \mathbf{D}I$ be càdlàg functions and $a > 0$. There exists some $\varepsilon < a$ such that $f \overset{\varepsilon}{\sim} g$ iff there exists some $\hat{\varepsilon} < a$ and partitions $\vec{s}^f, \vec{s}^g$, consisting of rational or discontinuity points for f and g respectively, witnessing that $f \overset{\hat{\varepsilon}}{\sim} g$.*

Proof

Take such $\varepsilon < a$ and a pair of partitions $\vec{t}^f, \vec{t}^g$ of the same length k such that $\left|t_i^f - t_i^g\right| \leq \varepsilon$ for all $1 \leq i \leq k$ and $|fx - gy| \leq \varepsilon$ for all $0 \leq i \leq k$ and $x \in [t_i^f, t_{i+1}^f)$ and $y \in [t_i^g, t_{i+1}^g)$. Find some ε' such that $\hat{\varepsilon} := \varepsilon + 2\varepsilon' < a$, and partitions $\vec{r}^f, \vec{r}^g$ that are ε'-tight and, since we may always refine an ε'-tight partition by adding more points, assume further that their segments have radius bounded by ε', and moreover contain at most one of the internal points in $\vec{t}^f$ or $\vec{t}^g$.

Each t_i^f belongs to some segment $[r_j^f, r_{j+1}^f)$, so define $s_i^f := r_j^f$. Since at most one t_i^f can belong to the same segment, we have that $\vec{s}^f$ is a partition of the interval. Moreover, each s_i^f is either rational or an ε'-discontinuity. Similarly, define $\vec{s}^g$. We have:

$$\left|s_i^f - s_i^g\right| \leq \varepsilon' + \left|t_i^f - t_i^g\right| + \varepsilon' \leq \varepsilon + 2\varepsilon' = \hat{\varepsilon}$$

Take $x \in [s_i^f, s_{i+1}^f)$ and $y \in [s_i^g, s_{i+1}^g)$. If $x \in [s_i^f, t_i^f)$, take $x' := t_i^f \in [t_i^f, t_{i+1}^f)$, and so by tightness, $|fx - fx'| \leq \varepsilon'$. If $x \notin [s_i^f, t_i^f)$, then $t_i^f \leq x < s_{i+1}^f \leq t_{i+1}^f$, and so taking $x' := x \in [t_i^f, t_{i+1}^f)$, we have $|fx - fx'| = 0 \leq \varepsilon'$, too. Similarly choose $y' \in [t_i^g, t_{i+1}^g)$ with $|gy - gy'| \leq \varepsilon'$. We have:

$$|fx - gy| \leq \varepsilon' + |fx' - gy'| + \varepsilon' \leq \varepsilon + 2\varepsilon' = \hat{\varepsilon}$$

Therefore, $\vec{s}^f$ and $\vec{s}^g$ witness that $f \overset{\hat{\varepsilon}}{\sim} g$. ■

This characterisation is measurable, in the sense that we can implement the predicate $\exists \varepsilon < a. f \overset{\varepsilon}{\sim} g$ as a Boolean predicate, as in Fig. 12. It detect two ε-similarity partitions consisting of ε'-discontinuities and rationals. Since there are only finitely many ε'-discontinuities, we can range over the countable set $\mathbf{List}\,(\mathbb{Q} \amalg \mathbb{N})$ which represents a putative interval partition whose internal points are either rationals or an index for an ε'-discontinuity of the corresponding function. The predicate $\varphi(\vec{a}, \vec{c})$ ensures that the indices in the list $\vec{a}$ indeed comprises of valid indices in the list $\vec{c}$. The partial function ψ then turns one of this representations into an actual list of real numbers.

$$
\begin{aligned}
&\left(\lambda(a,f,g).\exists\varepsilon.\varepsilon < a \wedge f \overset{\varepsilon}{\sim} g\right) : (0,\infty)\times(\mathbf{D}I)^2 \to \mathbb{B}\\
&\exists\varepsilon.\varepsilon < a \wedge f \overset{\varepsilon}{\sim} g := \exists\varepsilon,\varepsilon' \in (0,\infty)\cap\mathbb{Q}.\exists\vec{a}^f,\vec{a}^g \in \mathbf{List}\,(\mathbb{Q}\amalg\mathbb{N}).\mathbf{length}\,\vec{a}^f = \mathbf{length}\,\vec{a}^g\wedge\\
&\mathbf{let}\ n\ = \mathbf{length}\,\vec{a}^f\\
&\quad \vec{c}^f = \varepsilon'\text{-discontinuities}\,f\\
&\quad \vec{c}^g = \varepsilon'\text{-discontinuities}\,g\\
&\quad \varphi\ \ : \mathbf{List}\,(\mathbb{Q}\amalg\mathbb{N})\times\mathbf{List}\,\mathbb{R}\to\mathbb{B}\\
&\qquad = \lambda(\vec{a},\vec{c}).\mathbf{List.fold}\,(\lambda\,\{\mathbf{Nil}.\mathbf{true};(\iota_1 x) :- b.b;(\iota_2 i :- b).i < \mathbf{length}\,\vec{c}\wedge b\})\vec{a}\\
&\quad \psi\ \ : \mathbf{List}\,(\mathbb{Q}\amalg\mathbb{N}) \rightharpoonup \mathbf{List}\,\mathbb{R}\\
&\qquad = \lambda(\vec{a},\vec{c}).\mathbf{List.map}\,(\lambda\,\{\iota_1 q.q;\iota_2 i.c_i\})\vec{a}\\
&\mathbf{in}\ \varphi(\vec{a}^f,\vec{c}^f)\wedge\varphi(\vec{a}^g,\vec{c}^g)\wedge\\
&\mathbf{let}\ \vec{r}^f\ = \psi(\vec{a}^f,\vec{c}^f)\\
&\quad \vec{r}^g\ = \psi(\vec{a}^g,\vec{c}^g)\\
&\quad \vec{s}^f\ = a :- r^f -: b\\
&\quad \vec{s}^g\ = a :- r^g -: b\\
&\mathbf{in}\ \vec{r}^f,\vec{r}^g \in \mathbf{IPart}_I\wedge\\
&\forall i.1\le i\le n \Rightarrow \left|s_i^f - s_i^g\right| \le \varepsilon\wedge\\
&\forall i.0\le i\le n \Rightarrow \forall p,q\in\mathbb{Q}.p\in[s_i^f,s_{i+1}^f)\wedge q\in[s_i^g,s_{i+1}^g)\Rightarrow |fp-gq|\le\varepsilon
\end{aligned}
$$

Fig. 12. A simply-typed measurable implementation of the predicate $\exists\varepsilon.\varepsilon < a \wedge f \overset{\varepsilon}{\sim} g$

Its domain of definition contains those $(\vec{a},\vec{c})$ for which $\varphi(\vec{a},\vec{c})$ holds. Next, we require that our putative representations for partitions, $\vec{a}^f$, $\vec{a}^g$ do indeed satisfy φ, and we can therefore form the putative partitions $\vec{r}^f,\vec{r}^g : \mathbf{List}\,\mathbb{R}$, and require that they are partitions. Finally, we check that these partitions witness that $f \overset{\varepsilon}{\sim} g$, directly ensuring that corresponding internal points are ε-close, and indirectly via right-limits of rational sequences that images of points from corresponding segments are ε-close.

It follows therefore:

Lemma 12. *The Skorokhod metric is compatible with* $\mathbf{D}I$.

Proof

We have: $d_{\mathbf{S}}(f,g) := \inf\left\{a\in(0,\infty)\cap\mathbb{Q}\middle|\exists\varepsilon.\varepsilon < a\wedge f\overset{\varepsilon}{\sim} g\right\}$, measurable by type-checking. ■

Next, we turn to characterising when a sequence converges in in the Skorokhod topology $\mathcal{O}_{d_{\mathbf{S}}}$. We rephrase Kolmogorov's characterisation [1956, Thm. IV] in the language of ε-discontinuities. An *affine* partition of an interval is an interval partition with 0 or 1 internal points, hence two segments. We can refine an ε-tight affine partition to a tight one with exactly one internal point, which we call an *affinity* point. We have a partial selection function for such an affinity point:

$$
\begin{aligned}
&\text{-affPoint} : \{(\varepsilon,f,(x,y))|x<y\} \rightharpoonup \left\{t\in(x,y)\middle|\varepsilon\text{-tight}_{[x,y)}f[t]\right\}\\
&(\varepsilon,f,(x,y))\in\text{Dom-affPoint} := \mathbf{let}\ \vec{t} = \varepsilon\text{-discontinuities}\,f\ \mathbf{in}\\
&\qquad\qquad\left(\exists i.i<\mathbf{length}\,\vec{t},\varepsilon\text{-tight}_{[x,y)}f[t_i]\right)\vee\left(\exists t\in\mathbb{Q}.\varepsilon\text{-tight}_{[x,y)}f[t]\right)
\end{aligned}
$$

by first selecting the ε-discontinuity, if it exists, and otherwise selecting any rational affinity point.

Definition 13. *A sequence* $\vec{f}\in(\mathbf{D}I)^{\mathbb{N}}$ *is* convergent *when:*

- *it is Cauchy in the Skorokhod metric; and*

- *for all $\varepsilon > 0$ there is some $\delta > 0$ such that for all sub-intervals $[x, y) \subseteq I$ with $0 < |y - x| < \delta$, for every n, f_n has an ε-tight affine partition of $[x, y)$.*

We call such a δ as in the second condition an *ε-affinity threshold* for the sequence $\vec{f}$. The second condition implies that in every sub-interval whose length is below the ε-affinity threshold the sequence is uniformly affine ε-tight. The goal is to show that convergent sequences are those sequence that have a limit in the Skorokhod metric, that this limit is measurable given the sequence, and that being convergent is a Boolean predicate.

Lemma 14. *Let $\vec{f}$ be a sequence of càdlàg functions, and f be a càdlàg function. If $\lim \vec{f} = f$ in the Skorokhod topology, then $\vec{f}$ is convergent.*

Proof

The sequence is Cauchy by general principles. We prove the second condition. Assume $\vec{f} \to f$ in the Skorokhod topology. Since f is càdlàg, it has a $\frac{\varepsilon}{3}$-tight partition $\vec{u}$. Let δ_0 be the minimal segment length in $\vec{u}$:

$$\delta_0 \coloneqq \min_{i=0}^{\mathbf{length}\,\vec{u}} |u_{i+1} - u_i|$$

Consider a tail of the sequence whose components are distance $\frac{\varepsilon}{3}$ from f, i.e., for all $n \geq N$ we have $d_{\mathbf{S}}(f, f_n) < \min(\frac{\varepsilon}{3}, \frac{\delta_0}{5})$. For each $n < N$, we have a partition $\vec{v}^n$ in which f_n is ε-tight. Choose:

$$\delta \coloneqq \min \left\{\tfrac{\delta_0}{5}\right\} \cup \bigcup_{n<N} \bigcup_{i=0}^{\mathbf{length}\,\vec{v}^n} \left\{\left|v^n_{i+1} - v^n_i\right|\right\}$$

Take any interval $[x, y) \subseteq I$ such that $|y - x| < \delta$, and split into cases based on tail membership.

For $n < N$, we have that $x \in [v^n_i, v^n_{i+1})$ and $y \in [v^n_j, v^n_{j+1})$ for $i \leq j$. Assuming $i + 2 \leq j$ is absurd:

$$|y - x| \geq |t_{i+1} - t_{i+2}| \geq \delta > |y - x|$$

So either $i = j$ or $i = j + 1$. If $i = j$ then $[x, y) \subseteq [v^n_i, v^n_{i+1})$ is immediately tight, hence affinely so. If $i + 1 = j$, take $t \coloneqq v^n_{i+1}$. Then the partition $[t]$ has the two segments $[x, t) \subseteq [v^n_i, v^n_{i+1})$ and $[t, y) \subseteq [v^n_j, v^n_{j+1})$, and so f_n is tight in $[t]$.

For $n \geq N$, we know that $d_{\mathbf{S}}(f, f_n) < \frac{\delta}{5}$, and so there are partitions $\vec{t}^f, \vec{t}^n$ that witness $f \overset{\delta'}{\sim} f_n$ for some $\delta' < \delta$. Without loss of generality, we may assume $\left|t^n_i - t^n_{i+1}\right| \leq \delta$, as we can always refine each pair of corresponding segments by splitting them in half. We have that $x \in [t^n_i, t^n_{i+1})$ and $y \in [t^n_j, t^n_{j+1})$ with $i \leq j$. Calculate:

$$\left|t^f_i - t^f_{j+1}\right| \leq \left|t^f_i - t^n_i\right| + \left|t^n_i - x\right| + |x - y| + \left|y - t^n_{j+1}\right| + \left|t^n_{j+1} - t^f_{j+1}\right| < \delta' + 3\delta + \delta' \leq \delta_0$$

As a consequence either $[t^f_i, t^f_{j+1}] \subseteq [u_k, u_{k+1})$ or $[t^f_j, t^f_{j+1}] \subseteq [u_k, u_{k+2})$. Either way, we have a partition $[s]$ of $[t^f_{i+1}, t^f_j]$ in which f is $\frac{\varepsilon}{3}$-tight.

Consider the segment of s in $\vec{t}^f$, say $s \in [t^f_k, t^f_{k+1})$. Take the affinity point t to be the corresponding $t \coloneqq t^n_k$. We'll show that f_n is ε-tight in the partition $[t]$ of $[x, y)$. Take $p, q \in [x, t)$, say in the segments $p \in [t^n_\ell, t^n_{\ell+1})$ and $p \in [t^n_m, t^n_{m+1})$, and since $p, q < t = t^n_k$, we have $\ell, m < k$. Therefore $t^f_{i+1} \leq t^f_\ell, t^f_m < t^f_k \leq s$, and so $\left|f t^f_\ell - f t^f_m\right| \leq \frac{\varepsilon}{3}$. Calculate:

$$|f_n p - f_n q| \leq \left|f_n p - f t^f_\ell\right| + \left|f t^f_\ell - f t^f_m\right| + \left|f t^f_m - f_n q\right| \leq \varepsilon$$

Now take $p, q \in [t, y)$, say in the segments $p \in [t^n_\ell, t^n_{\ell+1})$, $q \in [t^n_m, t^n_{m+1})$. Since $p, q \geq t = t^n_k$, we have $\ell, m \geq k$. We need to take care in case one of $\ell = k$ or $m = k$, as then we should choose the

$$
\begin{array}{ll}
\delta : \mathbf{Convergent} \to (0,\infty)^{\mathbb{N}} & (\delta\vec{f})_n\vec{f} := \tfrac{1}{n}\text{-affThreshold}\vec{f} \\
\vec{k} : \mathbf{Convergent} \to \mathbf{Subseq} & \vec{k}(\vec{f}) := \mathbf{Stream.unfold} \\
& \qquad\qquad (\lambda(n,k).\mathbf{<}\begin{matrix}\text{head} : \text{Ƨ}\lambda k'.k' > k, \frac{b-a}{k'} < \min(\frac{1}{n}, \delta_n), \\ \text{tail} : (n+1, k')\end{matrix}\mathbf{>}) \\
& \qquad\qquad (0,0) \\
\vec{\vec{J}} : \mathbf{Convergent} \to \mathbf{Interval}^{\mathbb{N}} & \vec{J}_n := \left[(a + i\frac{(b-a)}{(2k\vec{f})_n}, a + (i+2)\frac{(b-a)}{(2k\vec{f})_n}) \middle| i = 0, \ldots, 2(k\vec{f})_n - 2\right] \\
\varphi : \mathbf{Convergent} \to I^{\mathbb{N}\times I} & (\varphi\vec{f})_n x := \mathbf{let}\,(u,v) = \text{Ƨ}(\vec{J}_n, \lambda(u,v).x \in (u,v))\ \mathbf{in}\ \tfrac{1}{n}\text{-affPoint}f_n[x,v) \\
\lim\vec{f} : \mathbf{Convergent} \to \mathbf{D}I & \lim\vec{f} := \lambda x.\lim_n t(\vec{f},x)_n
\end{array}
$$

Fig. 13. The limit in the Skorokhod topology

corresponding point in $[t^k_k, t^f_{k+1})$ to be s and not t^f_k, otherwise, we take it to be t^f_ℓ or t^f_m. Either way, the chosen points are then in $[s, t^f_{j+1})$. Conclude like for the other segment that $|f_n p - f_n q| \leq \varepsilon$. ■

Being convergent is almost measurable by type-checking. The gap is the need to range over all possible intervals $[x, y)$. However:

Lemma 15. *The convergent sequences are an event:* **Convergent** $\hookrightarrow (\mathbf{D}I)^{\mathbb{N}}$. *We have a selection principle for affinity thresholds:*

$$\text{-affThreshold} : \left\{\left(\varepsilon, \vec{f}\right) \in (0,\infty) \times \mathbf{Convergent}\right\} \to \left\{\delta \in (0,\infty) \middle| \delta \textit{ is an } \varepsilon\textit{-affinity threshold for } \vec{f}\right\}$$

Proof

Being Cauchy is an event because the Skorokhod metric is compatible.

A sequence $\vec{f}$ being tight in some affine partition of $J := [x, y)$ remains true when we move to a subinterval $J' \subseteq J$. As a consequence, we can strengthen the definition to use rational ε, δ, x and y without change. Then we read off the definition of the corresponding Boolean predicate and selection principle. ■

Fix a convergent sequence $\vec{f}$. We can measurably choose an affinity threshold δ_k for $\varepsilon_k := \frac{1}{k}$, uniformly for the whole sequence $\vec{f}$. Partitioning I into half-open segments of length below δ_k then gives us the data to construct the limit as in Fig. 13. Each point $x \in I$ lies in a sequence $J_k = [u_k, v_k)$ of segments, each of which has length below δ_n, and so there is a sequence of affinity points $t_n \to x$. As we will see, their images form a Cauchy sequence in $\mathbb{R}$, and so converges, and we may define the limit as $(\lim_n f_n)(x) := \lim_n f_n t_n$.

First, a few immediate observations:

Lemma 16. *The functions in Fig. 13 satisfy, for all $\vec{f} \in$* **Convergent***:*

- *For every n, the pairs of $\vec{J}_n$ make a list of open intervals that cover I and have length $\frac{b-a}{(k\vec{f})_n}$.*
- *For every $m \geq n$, the function f_m is affinely $\frac{1}{n}$-tight in every half-open subinterval of $\vec{J}_n$.*
- *For all $x \in I$, $(\varphi\vec{f})_n x \to x$.*

Proof

The first fact is by fiat, and note that the open intervals in $\vec{J}_n$ overlap. Since by definition we have that $\vec{k}$ is a monotone sequence, then $\frac{b-a}{(k\vec{f})_m} \leq \frac{b-a}{(k\vec{f})_n} < \delta_n$, and so f_m is affinely $\frac{1}{n}$-tight in every subinterval of $J_{n,i}$. Since x and $t_n := (\varphi\vec{f})_n x$ belong to the same interval in $\vec{J}_n$, we have $|x - t_n| \leq \frac{b-a}{(k\vec{f})_n} \leq \frac{1}{n} \to 0$ hence $t_n \to x$. ■

Next, we show that the limit is well-defined:

Lemma 17. *For every $\vec{f} \in$ **Convergent** and $x \in I$, the sequence $\left((\varphi \vec{f})_n x\right)_n$ is Cauchy.*

Proof
Take any $\vec{f}$, x and $\varepsilon > 0$. Choose k such that $\frac{1}{k} < \frac{\varepsilon}{3}$. Let $x \in J_k$ be the segment in $\vec{J}_k$ that x belongs to. ■

OK: Rest is dirty.

Since both conditions are Boolean predicates, we have a Borel embedding **Convergent** $\hookrightarrow (\mathbf{D}I)^{\mathbb{N}}$.

Lemma 18. *Let f, g be càdlàg functions. For all $\varepsilon > 0$ there exists $\delta_0 > 0$ such that for all $\delta \leq \delta_0$, if $\vec{t}^f, \vec{t}^g$ are partitions witnessing that $f \overset{\delta}{\sim} g$, then f is ε-tight in $\vec{t}^f$ and g is ε-tight in $\vec{t}^g$.*

Proof
Take $\delta_0 := \frac{\varepsilon}{2}$, $\delta \leq \delta_0$, $\vec{t}^f, \vec{t}^g$ partitions witnessing $f \overset{\varepsilon}{\sim} g$ and let $n :=$ **length** $\vec{t}^f =$ **length** t^g. Take any $0 \leq i \leq n$ and $x, y \in [\vec{t}^f_i, \vec{t}^f_{i+1})$, and then:

$$|fx - fy| \leq \left|fx - gt^g_i\right| + \left|gt^g_i - fy\right| \leq \delta + \delta \leq \varepsilon$$

and so f is ε-tight in $\vec{t}^f$. ■

The discontinuities in the sequence will converge to the discontinuities of the limit, but without knowing the limit exists, we do not even know how many such discontinuities we will have. Moreover, the number of discontinuities can fluctuate arbitrarily in a prefix of the sequence. The following lemma tells us it eventually stabilises:

Lemma 19. *Let $\vec{f}$ be a convergent sequence. Then for all $\varepsilon > 0$ there are some $n, N \in \mathbb{N}$ such that for all $m \geq N$ there are n distinct points $\vec{x}$ such that for all $\varepsilon' > \varepsilon$ and x, we have that x is an ε'-discontinuity of f_m iff it appears in $\vec{x}$, i.e.: $x = x_i$ for some i.*

Proof
■

We call n the *ε-discontinuity-threshold dimension*, or *ε-dimension* for short, and denote it by $\varepsilon\text{-dim}\,\vec{f}$. We can select the ε-dimension since its specification is an event in a countable discrete space, i.e., we have a measruable function:

$$\begin{aligned}
&\text{-dim} : (0, \infty) \times (\mathbf{D}I)^{\mathbb{N}} \to \mathbb{N} \\
&\varepsilon\text{-dim}\,\vec{f} = \text{℩}\,(\liminf_m \lambda n. \forall \varepsilon' \in \mathbb{Q}. \varepsilon' > \varepsilon \Rightarrow \mathbf{length}\ \varepsilon'\text{-discontinuities} f_m \leq n)
\end{aligned}$$

Let $\vec{f}$ be a convergent sequence with ε-dimension n. We can select (the earliest) tail in which all functions have n distinct ε-discontinuities, and thus we have selected a sequence in $\mathbb{R}^n$, which we call the *ε-dimensional tail sequence* whose i-th component is given measurably:

$$\begin{aligned}
&\text{-tail} : \left\{(\varepsilon, \vec{f}, i) \in (0, \infty) \times (\mathbf{D}I)^{\mathbb{N}} \times \mathbb{N} \middle| i < \varepsilon\text{-dim}\,\vec{f}\right\} \to \mathbb{R}^{\mathbb{N}} \\
&(\varepsilon\text{-tail}\vec{f})_i := \mathbf{let}\ N = \text{℩}\,(\forall m \geq N. n = \varepsilon\text{-discontinuities} f_m)_N\ \mathbf{in}\ ((\varepsilon\text{-discontinuities} f_m)_i)_{m \geq N}
\end{aligned}$$

Lemma 20. *The ε-dimensional tail sequence of $\vec{f}$ converges to a point in $\mathbb{R}^{\varepsilon\text{-dim}\,\vec{f}}$. If a component of an ε_1-dimensional tail sequence converges to the same point as a componenet of an ε_2-dimensional tail sequence, then the two sequences share a tail.*

Lemma 21. *A sequence of càdlàg functions $\vec{f} \in (\mathbf{D}I)^{\mathbb{N}}$ converges in the Skorokhod topology iff it is convergent. In that case, its limit is given by:*

$$\begin{aligned}\overset{d_S}{\lim} \vec{f} \coloneqq \lambda x.\mathbf{if}\ \exists \varepsilon \in \mathbb{Q}, i.x = \lim(\varepsilon\text{-tail}\vec{f})_i \\ \mathbf{then\ let}\ (\varepsilon, i) = \mathcal{Z}(\lambda(\varepsilon, i).x = \lim(\varepsilon\text{-tail}\vec{f})_i)\ \mathbf{in}\ \lim(\varepsilon\text{-tail}\vec{f})_i \\ \mathbf{else}\ \lim_n f_n x\end{aligned}$$

Putting everything together, we have:

Proposition 10. *The Skorokhod metric $d_{\mathbf{S}}$ is compatible with $\mathbf{D}I$, and has measurable limits. Since it is separable, it generates the events in the Skorokhod space: $\mathcal{B}_{d_S} = \mathcal{B}_{\mathbf{D}I}$.*

Proof

By Lemma 12, $d_{\mathbf{S}}$ is compatible with $\mathbf{D}I$. By Lemma 21, the event consisting of $d_{\mathbf{S}}$-converging is the event of convergent sequences, and the limit of a convergent sequence is measurable by type-checking. Since the Skorokhod metric is separable, we conclude by appeal to Proposition 5. ■

Received 1 May 2024