# A Monadic Theory of Point Processes




Swaraj Dash

Oriel College
University of Oxford

A thesis submitted for the degree of
*Doctor of Philosophy*
Michaelmas 2023

**Abstract**

A point process on a space is a random bag of elements of that space. In this thesis we explore programming with point processes in a monadic style. To this end we identify finite point processes on a space $X$ with probability measures on finite bags of elements in $X$. We describe this view of point processes using the composition of the Giry and bag monads on the category of measurable spaces and functions and prove that this composition also forms a monad using a distributive law for monads. We also explore distributive laws for infinite point processes, where an infinite point process on $X$ is a random stream of elements in $X$. We show that while there is no distributive law showing the space of infinite point processes to be a monad, there is a weak distributive law, which can be derived relatively generically. In the second half of this thesis we look at applications of the point process monad. We show that there is a morphism of monads giving the intensity measure of point processes and use it to derive a discrete version of Wald's lemma, a standard result from probability theory. Finally we use our point process monad to give a semantics to queries on probabilistic databases, which are databases with noisy data.

# Contents


**1 Introduction** **1**

1.1 What is a point process? . . . . . 1

1.2 Monads: what and why? . . . . . 2

1.3 Thesis outline and supporting publications . . . . . 3

**2 Preliminaries** **7**

2.1 Measure theory . . . . . 7

2.1.1 Lebesgue integration . . . . . 11

2.2 Monads in category theory . . . . . 13

2.2.1 Monad comprehensions . . . . . 14

2.2.2 Composing monads . . . . . 15

**3 Bags** **17**

3.1 Motivating measurable spaces of bags . . . . . 19

3.2 The measurable space of finite bags . . . . . 20

3.3 Measures on bags . . . . . 29

3.3.1 Defining a ring of sets of bags . . . . . 30

3.4 Measurable functions on bags . . . . . 35

3.4.1 Primitives for bag construction . . . . . 35

3.4.2 Measurable structural recursion on bags . . . . . 37

3.5 The bag monoid on measurable spaces . . . . . 45

3.6 The bag monad on measurable spaces . . . . . 46

3.7 The measurable space of finite bags as a quotient space . . . . . 49

3.7.1 The measurable space of finite bags is standard Borel . . . 50

3.8 Towards infinite bags . . . . . 52

**4 The point process monad** **59**

4.1 The Giry monad . . . 60
4.2 The point process endofunctor . . . 62
4.3 A monad for finite point processes . . . 63
4.3.1 A distributive law between the bag and Giry monads . . . 63
4.3.2 The distributive law as a measure . . . 70
4.3.3 The point process monad . . . 71
4.3.4 The unit and bind of the point process monad . . . 72
4.4 Monadic point process constructions . . . 75
4.4.1 Probability distributions as point processes . . . 76
4.4.2 Point process operations . . . 77
4.4.3 Constructing point processes . . . 78
4.5 Towards infinite point processes . . . 83
4.5.1 Measurable streams and the stream monad . . . 85
4.5.2 Non-existence of a distributive law . . . 87
4.5.3 A weak distributive law . . . 91
4.6 Summary and related work . . . 102

**5 The intensity measure 105**
5.1 The monad of all measures . . . 106
5.2 The intensity measure as a monad morphism . . . 107
5.2.1 Bags as measures . . . 107
5.2.2 The intensity measure morphism . . . 109
5.2.3 Intensity as a monad morphism . . . 110
5.3 Applications of the monad morphism . . . 113
5.3.1 Intensity of a Poisson point process . . . 113
5.3.2 Discrete Wald's lemma . . . 114
5.4 Related work . . . 116

**6 Probabilistic databases as finite point processes 117**
6.1 An algebraic query language for bag databases . . . 118
6.1.1 Measurability of Balg queries . . . 120
6.1.2 Grouping and aggregation . . . 123
6.2 Generating probabilistic databases . . . 125
6.2.1 Randomising attributes . . . 125
6.2.2 Adding random records . . . 126
6.3 Related work . . . 127

**7 Conclusion** **129**
7.1 Future work . . . . . 129
7.2 Takeaways . . . . . 130

**Bibliography** **133**

**A Monadic point process constructions in LazyPPL** **139**

# Chapter 1

# Introduction

A point process is a random bag. Similarly to how a draw from a probability distribution on a set $X$ is a random element from $X$, a draw from a point process on $X$ is a random bag (also known as multiset) of elements from $X$. There are two layers of randomness in point processes – in addition to the individual elements in a bag drawn from a point process being random, the quantity of these elements is itself a random number. In this thesis we study point processes through a functional programming and category theoretic lens. In particular, we look at the monadic structure on point processes induced by the monadic structure of probability distributions and the monoidal structure on bags. With this idea we are able to easily define a rich family of transformations on point processes, even in the case when $X$ is an uncountable set, which is where issues regarding measurability would normally arise.

## 1.1 What is a point process?

Point processes (e.g. [Daley and Vere-Jones, 2006]) are random collections of points, and more precisely, random bags of points. They serve as important tools in statistical analysis, where they are used to study various phenomena in fields as diverse as ecology, astronomy, computational neuroscience, and telecommunications, and in Bayesian analysis, where they are used for probabilistic inference (e.g. [Herlau et al., 2016]).

Let us first better understand what point processes are, starting with a finite example.

In ecological studies a simple instrument for studying the spatial distribution of plants is a quadrat, a rectangular frame subdivided into smaller rectangles. An easy way to determine how certain plans are distributed on land is by placing these quadrats on the ground and counting the number of various species in each smaller rectangle. Taking such measurements over a large area we can get a useful sample of the average locations of the plants we are studying. Seeing the sub-rectangles as a finite set of points, we get a distribution of the number of plants in each region, which can be seen as a bag of points on our space.

In particular, this defines a point process on our finite space, where each draw corresponds to a bag of points. The study of point processes takes this to the next level by analysing spatial distributions on continuous uncountable spaces, such as the distribution of stars in the sky, or the locations of trees in forests.

In this thesis we look at point processes as objects in the category of measurable spaces, and study them abstractly using monads. We formally introduce them in Chapter 4 and look at examples in Section 4.4.

## 1.2 Monads: what and why?

The concept of monads originally comes from category theory and has proved to be an important tool in the areas of programming language semantics as well as functional programming for several decades now. A first introduction to monads is by analogising them as containers of values. There then arise two operations which can be performed in a natural way: placing single objects into containers, and condensing containers of containers into a single container. Furthermore these operations need to obey some algebraic laws, which ensures that programs written in a monadic style can be safely transformed and reasoned about, and written in a generic manner.

In general, monads are not strictly limited to container-like situations. In functional programming especially, monads encapsulate the notion of sequencing together effectful impure computations. In this work we are especially concerned with two monads, the bag monad and the monad on probability distributions. By writing programs in these monads we can make use of abstract algebraic properties to build together complex programs. In particular, by using the probability and bag monads, and showing that their composition is also a monad, we can define useful

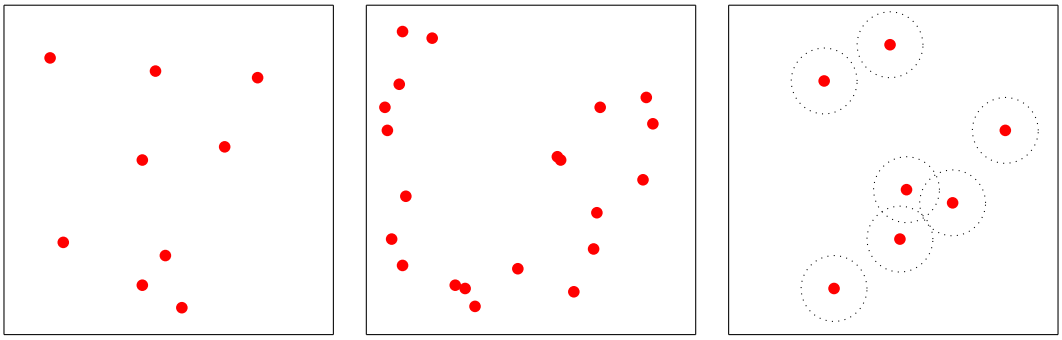


Figure 1.1: Samples from various point processes defined using the point process monad in LazyPPL.

statistical models as point processes, which is the main subject of this thesis. A formal definition of monads, along with related topics, is present in Section 2.2.

As a quick preview of our work, we show how the point process monad allows us to define many interesting and complicated point processes concisely and measurably. Figure 1.1 contains examples of three such point processes, namely

1. the Poisson point process, where the locations of points are not correlated,
2. the Thomas cluster process, where points tend to cluster together, and
3. the Matern hard-core point process, where points exhibit repulsions.

We explain these point processes in detail in Chapter 4 and refer the reader to Appendix A to see the implementation.

## 1.3 Thesis outline and supporting publications

The work in this thesis is primarily based joint work with Sam Staton published in the following two papers.

1. Swaraj Dash & Sam Staton. **A monad for probabilistic point processes.** In *Proceedings of Applied Category Theory 2020.*
2. Swaraj Dash & Sam Staton. **Monads for measurable queries in probabilistic databases.** In $37^{th}$ *Conference on Mathematical Foundations of Programming Semantics.*

We also make use of a Haskell library (LazyPPL) written as part of a third publication,

jointly written with Younesse Kaddar, Hugo Paquet, and Sam Staton.

3. Swaraj Dash, Younesse Kaddar, Hugo Paquet, Sam Staton. **Affine monads and lazy structures for Bayesian programming.** In *Proceedings of* $50^{th}$ *ACM SIGPLAN Symposium on Principles of Programming Languages.*

Below we give an outline of this thesis. Note that it combines and generalises the results from [Dash and Staton, 2020] and [Dash and Staton, 2021]. In particular, the material in Chapters 4 and 5 is based on [Dash and Staton, 2020] and the material in Chapters 3 and 6 primarily comes from [Dash and Staton, 2021].

**Chapter 2.** In this chapter we recall background material used in the rest of this thesis. In the first half of this chapter we provide an overview of the measure theoretic definitions and results relevant to our work. This includes standard constructions in measure theory, such as products and coproducts, and Lebesgue integration, a formalism of integration applied to measurable spaces, which we use for calculating on measurable spaces. Next, we go over the topic of monads, and in particular the syntax of monad comprehensions, which are our preferred syntactic construct for monadic expressions. Finally we talk about the problem of composing monads, and how to do so using distributive laws between monads so that composite endofunctors exhibit monadic structure.

**Chapter 3.** Here we define the measurable space of bags and show how to define measures on measurable spaces of bags in Sections 3.1 - 3.3. In Section 3.4 we give a fold construct for defining structurally recursive functions on bags and prove that the fold of measurable functions is measurable. By folding appropriate functions we prove that the measurable space of bags is the free commutative monoid in Section 3.5. In Section 3.6 we make use of the free commutative monoid structure to give a monad on the measurable space of bags. Finally we prove that our space of bags is quotient measurable space in Section 3.7 and briefly discuss the possibility of extending this to infinite bags in Section 3.8.

**Chapter 4.** Having constructed the bag monad $\mathbf{B}$ on measurable spaces, we define the point process endofunctor $\mathbf{G} \circ \mathbf{B}$ as the composition of the Giry monad $\mathbf{G}$ and the bag monad $\mathbf{B}$ in Section 4.2. In Section 4.3 we define a natural transformation $\mathbf{BG} \to \mathbf{GB}$ using our fold construct and prove that it is a distributive law of monads, showing that $\mathbf{G} \circ \mathbf{B}$ is a monad. *This is the main result of the thesis.* We use this

monad in Section 4.4 to present various measurable constructions studied in point process theory, showing that they can be defined concisely using the point process monad and the bag monoid. In Section 4.5 we look at the stream endofunctor $\mathbf{S}$ on measurable spaces and show that it is a strong monad. We then prove that there is no distributive law $\mathbf{SG} \to \mathbf{GS}$ but that there is a very natural weak distributive law between the Giry and stream monads.

**Chapter 5.** This chapter is about the intensity measure of point processes, which is a widely used concept in characterising point processes. In Section 5.2 we show that there is a natural monad morphism from the point process monad to the monad of all measures $\mathbf{M}$ by injecting both $\mathbf{G}$ and $\mathbf{B}$ into $\mathbf{M}$. We show how this monad morphism can be used to compute the intensity measures of point processes in Section 5.3 with the example of a Poisson point process. Finally we show that a discrete version of Wald's lemma, a standard result in probability theory, falls out as a special case of our monad morphism.

**Chapter 6.** In this final chapter, by using the results of the previous chapters, we give an immediate application of the point process monad by giving a semantic account of queries on probabilistic databases. Probabilistic databases are databases modelling noisy data, where individual attributes can come from probability distributions and where entire rows of databases have probabilities of occuring. In this way probabilistic databases are examples of point processes. In Section 6.1 we introduce a query language on bags, extend it by adding grouping and aggregation operations, and show that this language lifts to the measurable setting, allowing for measurable queries on probabilistic databases. We show that the measurability results for probabilistic databases, which are the main results of Grohe and Lindner [2019], follow immediately from our monadic methods. In Section 6.2 we discuss constructing such databases, by showing how the point process monad can be used to generate noise in individual attributes and entire rows.

# Chapter 2

# Preliminaries

In this chapter we review all the necessary mathematical preliminaries needed for our work, starting with an overview of the measure theoretic definitions and results we use throughout. All the relevant definitions (and more) can be found in standard measure theory textbooks [Fremlin, 2000, Halmos, 1976]. Next we review the theory of monads, including useful syntax for working with them, and how to compose them using what are known as distributive laws.

## 2.1 Measure theory

Measure theory is a branch of mathematics that deals with the concept of measure, which is a way of assigning a size or quantity to certain sets of objects. It is a fundamental tool in modern mathematics, with applications in fields such as probability, real analysis, and functional analysis.

One of the main roles of measure theory is to provide a foundation for probability theory on uncountable spaces, such as the real numbers. In probability theory, we often deal with random variables that take on values in some set of outcomes, known as the sample space. When the sample space is finite or countable, it is possible to simply assign individual probabilities to individual outcomes of the sample space. This fails when we consider uncountable spaces, such as the set of reals, where it is impossible to construct such an assignment to each real number. What is possible, however, is to assign probabilities to sets of real numbers. In order to define a probability measure on an uncountable sample space we assign

a probability to subsets of the sample space such that these probabilities satisfies certain usual axioms. For example, in the uniform distribution on the unit interval the probability of sampling the number 0.4 is zero (i.e. the singleton set 0.4 has measure zero), but the probability of sampling around the point 0.4 within a distance of, say, 0.05 is 1/10 (i.e. the interval $[0.35, 0.45]$ is 1/10). This allows us to perform probabilistic reasoning on uncountable spaces and make predictions about random phenomena in a rigorous and consistent way.

We begin with stating all the measure-theoretic definitions and results we make use of in the rest of this work. First, we define a $\sigma$-algebra, the set of subsets of the sample space of interest. It is precisely the elements of the $\sigma$-algebra, subsets of our sample space, to which we will assign a measure.

**Definition 2.1.1.** A *$\sigma$-algebra* on a set $X$ is a family $\Sigma_X$ of subsets of $X$ containing $\varnothing$ and $X$ that is closed under complements and countable unions. The pair $(X, \Sigma_X)$ is called a *measurable space* (we just write $X$ when $\Sigma_X$ can be inferred from context).

**Definition 2.1.2.** A measure $\nu : \Sigma_X \to \mathbb{R}^+_\infty$ is a function that maps subsets $U \in \Sigma_X$ of a set $X$ to the extended real line $\mathbb{R}^+_\infty = [0, +\infty]$, satisfying the following properties:

1. $\mu(\emptyset) = 0$.
2. For all countable collections of disjoint sets $A_i \in \Sigma_X$, $\nu\left(\bigcup_i A_i\right) = \sum_i \nu(A_i)$.

$\nu$ is called a *probability measure* if $\nu(X) = 1$.

These properties are known as the $\sigma$-additivity or countable additivity of the measure. The first property states that the measure of the empty set is zero, and the second property states that the measure of a countable union of disjoint sets is the sum of the measures of the individual sets.

In addition, a measure is said to be finite if $\mu(X) < \infty$, where $X$ is the underlying set of the measure. The measure $\mu$ is called $\sigma$-finite if $X$ is a countable union of measurable sets each with finite measure. In particular all probability measures are $\sigma$-finite.

**Definition 2.1.3** (Generated $\sigma$-algebra)**.** Let $\mathcal{F}$ be an arbitrary family of subsets of $X$. The intersection of a collection of $\sigma$-algebras is also a $\sigma$-algebra, and so we define the $\sigma$-algebra generated by $\mathcal{F}$ to be the intersection of all the $\sigma$-algebras

containing $\mathcal{F}$. This is then the largest $\sigma$-algebra containing $\mathcal{F}$ and is denoted $\sigma(\mathcal{F})$.

In general the generation of $\sigma$-algebras is a transfinite process, because of the requirement of having to be closed under both countable unions and intersections, but the definition above is easier to work with.

**Example 2.1.4.** Let $X$ be any set. $\Sigma_X$ is called the indiscrete (or trivial) $\sigma$-algebra when $\Sigma_X = \{\varnothing, X\}$ and the discrete (or powerset) $\sigma$-algebra when $\Sigma_X = \mathcal{P}(X)$.

**Definition 2.1.5.** The *Borel sets* form the least collection $\Sigma_{\mathbb{R}}$ of subsets of $\mathbb{R}$ containing intervals $(a, b) \subseteq \mathbb{R}$ which is closed under complementation and countable unions, making $(\mathbb{R}, \Sigma_{\mathbb{R}})$ a measurable space on the reals. For a general topological space $X$, a Borel is any set that can be formed from open sets through the operations of countable union, countable intersection, and relative complement. The Borel $\sigma$-algebra $\mathcal{B}(X)$ is then the $\sigma$-algebra generated by the open sets of $X$.

Next we define what constitutes a morphism between two measurable spaces. A measurable function between two measurable spaces is a function between the two underlying sets such that it preserves the structure of the $\sigma$-algebra on the domain and codomain. Intuitively, this means that the function does not "distort" the measurable space in an inconsistent way.

**Definition 2.1.6.** Let $(X, \Sigma_X)$ and $(Y, \Sigma_Y)$ be measurable spaces. A measurable function $f : (X, \Sigma_X) \to (Y, \Sigma_Y)$ is a function $f : X \to Y$ such that the inverse image map $f^{-1}[U] = \{x \in X \mid f(x) \in U\} \in \Sigma_X$ when $U \in \Sigma_Y$.

This criterion for measurability presents a big proof burden, since in order to check if a function is measurable we need to consider the inverse image map for every measurable set in the codomain. In the uncountable case this is difficult because of the transfinite construction of $\sigma$-algebras, meaning it does not simply suffice to check measurability on some subset of the $\sigma$-algebra.

It is not uncommon, however, to normally consider $\sigma$-algebras generated by some family of subsets. In this case there is a standard result which allows us to prove the measurability of a function just by checking the measurability condition on the elements of the generating family. The proof relies on the fact that unions and intersections preserve inverse images, i.e., $f^{-1}[X \cup Y] = f^{-1}[X] \cup f^{-1}[Y]$ and similarly for intersection.

**Lemma 2.1.7.** *Let $(X, \Sigma_X)$ and $(Y, \sigma(\mathcal{F}))$ be two measurable spaces, where the*

*$\sigma$-algebra for $Y$ is generated by $\mathcal{F}$. Let $f : X \to Y$ be an arbitrary function. Then $f$ is a measurable function if $\forall U \in \mathcal{F}.\ f^{-1}[U] \in \Sigma_X$.*

**Definition 2.1.8.** Let $(X, \Sigma_X)$ and $(Y, \Sigma_Y)$ be two measurable spaces, $f : X \to Y$ a measurable function, and $\nu : \Sigma_X \to \mathbb{R}^+_\infty$ be a measure on $X$. Then the pushforward of $\nu$ is a measure on $Y$ given by $\nu \circ f^{-1} : \Sigma_Y \to \mathbb{R}^+_\infty$.

The category of measurable spaces has countable products and coproducts. For products we first discuss the finite case. Given two measurable spaces $(X, \Sigma_X)$ and $(Y, \Sigma_Y)$, their product is given by the measurable space $(X \times Y, \Sigma_{X\times Y})$ where the $\sigma$-algebra for this measurable space is generated by the set containing all the rectangles formed from $\Sigma_X$ and $\Sigma_Y$.

$$\Sigma_{X\times Y} = \sigma(\{U \times V \mid U \in \Sigma_X, V \in \Sigma_Y\})$$

Categorically speaking, this is the smallest $\sigma$-algebra such that the projection maps $\pi_1 : X \times Y \to X$ and $\pi_2 : X \times Y \to Y$ are measurable. When $\nu_1 : \Sigma_X \to \mathbb{R}^+_\infty$ and $\nu_2 : \Sigma_Y \to \mathbb{R}^+_\infty$ are measures on $X$ and $Y$ respectively, the measure $\nu_1 \otimes \nu_2 : \Sigma_{X\times Y} \to \mathbb{R}^+_\infty$ is called a *product measure* on $X \times Y$. The product measure satisfies the property that $(\nu_1 \otimes \nu_2)(U \times V) = \nu_1(U)\nu_2(V)$ for all $U \in \Sigma_X$ and $V \in \Sigma_Y$. One way to define a product measure is by defining a map on all $U \times V$ ($U \in \Sigma_X, V \in \Sigma_Y$) satisfying this property and then extending this map to the entire $\sigma$-algebra $\Sigma_{X\times Y}$. A well-known theorem in measure theory which can be used for defining such a (unique) extension is the Carathéodory extension theorem, which we state and make use of in Chapter 3.

In the countable case, given a countable collection of measurable spaces $\{(X_i, \Sigma_{X_i})\}_{i\in\mathbb{N}}$, the product measurable space is $(\prod_{i\in\mathbb{N}} X_i, \Sigma_{\prod_{i\in\mathbb{N}} X_i})$ where the $\sigma$-algebra $\Sigma_{\prod_{i\in\mathbb{N}} X_i}$ is generated by cylinders instead. These are measurable sets built from the product of other measurable sets where all but finitely many of the terms are the entire space. These sets can be written concisesly as the product of $n$ measurable sets $\prod_{i=1}^n U_i$ (where $U_i \in \Sigma_{X_i}$) for some $n \in \mathbb{N}$ with the rest of the sets being the entire respective spaces $\prod_{i=n+1}^\infty X_i$.

$$\Sigma_{\prod_{i\in\mathbb{N}} X_i} = \sigma\left(\left\{\prod_{i=1}^{n} U_i \times \prod_{i=n+1}^{\infty} X_i \;\middle|\; U_i \in \Sigma_{X_i}, n \in \mathbb{N}\right\}\right)$$

Given a collection of measurable spaces $\{(X_i, \Sigma_{X_i})\}_{i\in\mathbb{N}}$ we define their coproduct to be the measurable space $(\bigsqcup_{i\in\mathbb{N}} X_i, \Sigma_{\bigsqcup_{i\in\mathbb{N}} X_i})$ where the $\sigma$-algebra is defined to be the smallest $\sigma$-algebra such that all the injection maps $\mathsf{inj}_i : X_i \to \bigsqcup_{i\in\mathbb{N}} X_i$ are measurable. There is an explicit description of this $\sigma$-algebra when all the sets $X_i$ are disjoint. Then a subset $S \subseteq \bigsqcup_{i\in\mathbb{N}} X_i$ is measurable exactly if $S \cap X_i$ is a measurable subset of $X_i$ for each $i \in \mathbb{N}$.

$$\Sigma_{\bigsqcup_{i\in\mathbb{N}} X_i} = \left\{ S \subseteq \bigsqcup_{i\in\mathbb{N}} X_i \;\middle|\; S \cap X_i \in \Sigma_{X_i} \right\}$$

Products distribute over sums in the category of measurable spaces. This can be seen by noting that the set-theoretic bijection $X \times (Y \sqcup Z) \to X \times Y \sqcup X \times Z$ is also a measurable map, making it an isomorphism in **Meas**.

**Definition 2.1.9.** A measurable space $(X, \Sigma_X)$ is said to be standard Borel if there exists a metric on $X$ that makes it a complete separable metric space in such a way that $\Sigma_X$ is then the Borel $\sigma$-algebra.

**Theorem 2.1.10** (Kuratowski)**.** *Let $X$ be a Polish space (a separable completely metrizable topological space). Then $X$ as a Borel space is measurably isomorphic to either $(\mathbb{R}, \Sigma_{\mathbb{R}})$ or is it countable and discrete.*

Standard Borel spaces are very well behaved, such as being closed under countable products and countable coproducts. For example, the equality predicate $X \times X \to$ Bool is measurable when $X$ is a standard Borel space.

### 2.1.1 Lebesgue integration

Lebesgue integration is a theory of integration with respect to measures. In the following let $(X, \Sigma_X)$ be a measurable space and $\nu : \Sigma_X \to \mathbb{R}^+_\infty$ a measure on it.

The indicator function of a subset $A$ of $X$ is the following function.

$$\mathbf{1}_A : X \to \{0, 1\}$$

$$\mathbf{1}_A(x) = \begin{cases} 1, \text{ if } x \in A \\ 0, \text{ otherwise} \end{cases}$$

**Definition 2.1.11** (Simple function)**.** A function $f : X \to \mathbb{R}$ is called a simple measurable function if it is the linear combination of indicator functions $\sum_k a_k \mathbf{1}_{S_k}$ where the coefficients $a_k$ are positive real numbers and $S_k$ are disjoint measurable subsets of $X$.

When $E$ is a measurable subset of $X$ and $s$ a measurable simple function, we define the integral of $s$ with respect to $\nu$ as the following sum.

$$\int_E s \, \mathrm{d}\nu = \int_E \sum_k a_k \mathbf{1}_{S_k} \, \mathrm{d}\nu = \sum_k a_k \nu(S_k \cap E)$$

**Definition 2.1.12** (Lebesgue integral)**.** Let $f : X \to \mathbb{R}^+_\infty$. Then we define the integral of $f$ with respect to the measure $\nu$ on a measurable subset $E$ in terms of the following supremum of integrals of simple functions.

$$\int_E f \, \mathrm{d}\nu = \sup \left\{ \int_E s \, \mathrm{d}\nu \;\middle|\; 0 \leq s \leq f, s \text{ is simple} \right\}$$

A note on notation. It is common to write $\int_E f \, \mathrm{d}\nu$ for the integral of $f$ with respect to $\nu$. The following are equivalent notations for writing this same integral, but where the variable with respect to which we integrate is made explicit.

$$\int_E f \, \mathrm{d}\nu = \int_{x \in E} f(x) \, \mathrm{d}\nu(x) = \int_{x \in E} f(x) \nu(\mathrm{d}x)$$

There is a generalisation of Fubini's theorem (a theorem from analysis which gives conditions under which it is possible to compute a double integral by using iterated integrals) to Lebesgue integration, known as the Fubini-Tonelli theorem.

**Theorem 2.1.13** (Fubini-Tonelli)**.** *Let $(X, \Sigma_X)$ and $(Y, \Sigma_Y)$ be measurable spaces and $\nu_1 : \Sigma_X \to \mathbb{R}^+_\infty$ and $\nu_2 : \Sigma_Y \to \mathbb{R}^+_\infty$ be $\sigma$-finite measures. Let $f : X \times Y \to \mathbb{R}^+_\infty$ be a non-negative measurable function. Then the following integrals are equal.*

$$\begin{aligned} \int_{X \times Y} f(x,y)(\nu_1 \otimes \nu_2)(\mathrm{d}(x,y)) &= \int_X \left( \int_Y f(x,y) \nu_2(\mathrm{d}y) \right) \nu_1(\mathrm{d}x) \\ &= \int_Y \left( \int_X f(x,y) \nu_1(\mathrm{d}x) \right) \nu_2(\mathrm{d}y) \end{aligned}$$

Finally there is a change-of-variables formula for Lebesgue integration making it possible to express integrals with respect to pushforward measures in terms of integrals on the original measures. It is similar to the usual change-of-variables formula from high-school calculus.

**Theorem 2.1.14** (Change-of-variables formula)**.** *Let $(X_1, \Sigma_{X_1})$ and $(X_2, \Sigma_{X_2})$ be measurable spaces and let $\nu : \Sigma_{X_1} \to \mathbb{R}^+_\infty$ be a measure on $X_1$ and $f : X_1 \to X_2$ be a measurable map. Let $\nu \circ f^{-1} : \Sigma_{X_2} \to \mathbb{R}^+_\infty$ be the pushforward measure induced by $f$. Finally let $g : X_2 \to Y$ be any measurable function. Then the following integrals are equal.*

$$\int_{X_2} g \, \mathrm{d}(\nu \circ f^{-1}) = \int_{X_1} g \circ f \, \mathrm{d}\nu$$

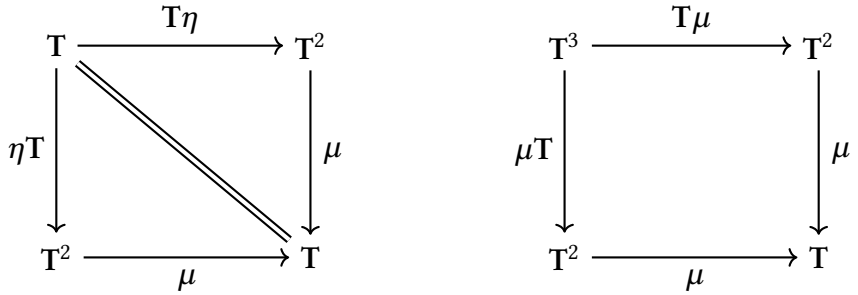
*Writing this out with explicit integration variables we get the following.*

$$\int_{x_2 \in X_2} g(x_2) \; (\nu \circ f^{-1})(\mathrm{d}x_2) = \int_{x_1 \in X_1} (g \circ f)(x_1) \; \nu(\mathrm{d}x_1)$$

## 2.2 Monads in category theory

Monads are an important concept in category theory, and they can be used to model many different kinds of computation, including things like side effects, exceptions, and nondeterminism, and randomness [Moggi, 1991]. They provide a framework for understanding how different computations can be related, transformed, and composed, and they are an essential tool for working with complex mathematical structures in a principled manner.

**Definition 2.2.1.** Let $C$ be a category. Then a monad on $C$ is defined to be a triple $(\mathbf{T} : C \to C, \eta : 1 \to \mathbf{T}, \mu : \mathbf{T}^2 \to \mathbf{T})$ consisting of an endofunctor on $C$ and two natural transformations that make the following diagrams commute.



**Definition 2.2.2.** Given a monoidal category $(C, \times : C \times C \to C, I)$ we say $\mathbf{T} : C \to C$ is a strong monad if there exists a natural transformation $t_{X,Y} : X \times \mathbf{T}Y \to \mathbf{T}(X \times Y)$, called the strength, satisfying certain coherence axioms (refer to [Kock, 1970] for the full set of coherence axioms).

**Definition 2.2.3.** When $C$ is a symmetric monoidal category with swap map $s_{X,Y} : X \times Y \to Y \times X$, we can construct the costrength $t'_{X,Y} : \mathbf{T}X \times Y \to \mathbf{T}(X \times Y)$.

$$t'_{X,Y} = \mathbf{T}X \times Y \xrightarrow{s_{\mathbf{T}X,Y}} Y \times \mathbf{T}X \xrightarrow{t_{X,Y}} \mathbf{T}(Y \times X) \xrightarrow{\mathbf{T}s_{Y,X}} \mathbf{T}(X \times Y)$$

We say $\mathbf{T}$ is a commutative monad if it makes the following diagram commute.

$$\begin{array}{ccccc}
\mathbf{T}X \times \mathbf{T}Y & & \xrightarrow{t_{\mathbf{T}X,Y}} & & \mathbf{T}(\mathbf{T}X \times Y) \xrightarrow{\mathbf{T}t'_{X,Y}} \mathbf{T}^2(X \times Y) \\
\Big\downarrow{\scriptstyle t'_{X,\mathbf{T}Y}} & & & & \Big\downarrow{\scriptstyle \mu_{X\times Y}} \\
\mathbf{T}(X \times \mathbf{T}Y) & \xrightarrow[\mathbf{T}t_{X,Y}]{} & \mathbf{T}^2(X \times Y) & \xrightarrow[\mu_{X\times Y}]{} & \mathbf{T}(X \times Y)
\end{array}$$

**Definition 2.2.4.** There is a category $C_{\mathbf{T}}$, called the Kleisli category, associated to any monad $(\mathbf{T} : C \to C, \eta : 1 \to \mathbf{T}, \mu : \mathbf{T}^2 \to \mathbf{T})$. The objects of this category are the same as the objects of the base category $C$, and the morphisms $f : X \to Y$ are morphisms $f : X \to \mathbf{T}Y$ in $C$. The composition of two morphisms $f : X \to Y$ and $g : Y \to Z$ in $C_{\mathbf{T}}$ is given by the following composition in $C$.

$$g \circ_{\mathbf{T}} f = X \xrightarrow{f} \mathbf{T}Y \xrightarrow{\mathbf{T}g} \mathbf{T}^2 Z \xrightarrow{\mu_Z} \mathbf{T}Z : X \to \mathbf{T}Z$$

### 2.2.1 Monad comprehensions

Let $C$ be a symmetric monoidal category and let $\mathbf{T} : C \to C$ be a strong monad on it. When programming with monads it is often convenient to focus on Kleisli composition in a stylised form, using the function $\gg\!=_{\mathbf{T}} : C(X, \mathbf{T}Y) \to C(\mathbf{T}X, \mathbf{T}Y)$ (pronounced *bind*). We write $\alpha \gg\!=_{\mathbf{T}} f$ for $\gg\!=_{\mathbf{T}} (f)(\alpha)$.

There exists convenient syntactic sugar when needing to write expressions consisting purely of the monadic unit and bind, known as monad comprehension notation, due to Wadler [1992].

Given $f_1 : A \to \mathbf{T}X_1$, $f_2 : A \times X_1 \to \mathbf{T}X_2$, …, $f_n : A \times X_1 \times \cdots \times X_{n-1} \to \mathbf{T}X_n$, and

given $g : A \times X_1 \times X_2 \times \cdots \times X_n \to Y$, we write

$$\begin{aligned} a \;\mapsto\; \{\!\!\{ g(a, x_1, \ldots x_n) \mid\; & x_1 \leftarrow f_1(a), \\ & x_2 \leftarrow f_2(a, x_1), \\ & \ldots, \\ & x_n \leftarrow f_n(a, x_1, \ldots, x_{n-1}) \}\!\!\}_{\mathbf{T}} \end{aligned}$$

for the composite morphism

$$A \xrightarrow{\overline{f_1}} \mathbf{T}(A \times X_1) \xrightarrow{\gg\!= \overline{f}_2} \mathbf{T}(A \times X_1 \times X_2) \to \ldots \xrightarrow{\gg\!= \overline{f}_n} \mathbf{T}(A \times X_1 \times X_2 \times \cdots \times X_n) \xrightarrow{\mathbf{T}(g)} \mathbf{T}(Y)$$

where

$$\overline{f_i} = A \times X_1 \times \ldots X_{i-1} \xrightarrow{(\mathrm{id}, f_i)} A \times X_1 \times \ldots X_{i-1} \times \mathbf{T}X_i \xrightarrow{\mathrm{str}} \mathbf{T}(A \times X_1 \times \cdots \times X_i).$$

We make extensive use of monad comprehensions in this work due to their conciseness and the additional visual intuition that they provide when working with monads. We will often simplify our monadic computations using the monad and functor laws. These simplifications are listed below.

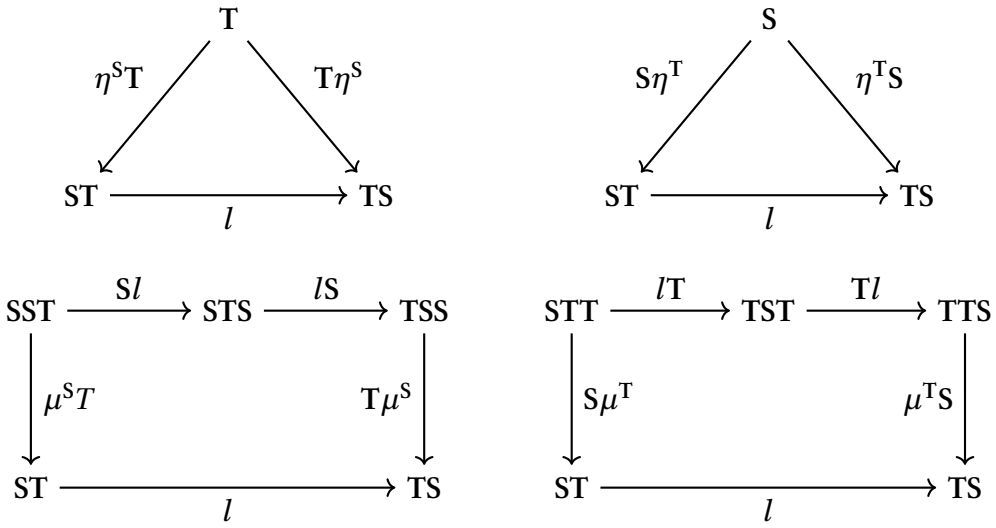
- $\{\!\!\{ x \mid - \}\!\!\}_{\mathbf{T}} = \eta(x) = \{\!\!\{ y \mid y \leftarrow \eta(x) \}\!\!\}_{\mathbf{T}}$
- $\mathbf{T}(f)\{\!\!\{ x \mid x \leftarrow \alpha \}\!\!\}_{\mathbf{T}} = \{\!\!\{ f(x) \mid x \leftarrow \alpha \}\!\!\}_{\mathbf{T}}$
- $\{\!\!\{ g(y) \mid y \leftarrow \{\!\!\{ f(x) \mid x \leftarrow \alpha \}\!\!\}_{\mathbf{T}} \}\!\!\}_{\mathbf{T}} = \{\!\!\{ g(f(x)) \mid x \leftarrow \alpha \}\!\!\}_{\mathbf{T}}$
- $\mu\{\!\!\{ \{\!\!\{ f(y) \mid y \leftarrow \alpha \}\!\!\}_{\mathbf{T}} \mid \alpha \leftarrow \beta \}\!\!\}_{\mathbf{T}} = \{\!\!\{ f(y) \mid \alpha \leftarrow \beta, y \leftarrow \alpha \}\!\!\}_{\mathbf{T}}$

When $\mathbf{T}$ is the powerset monad on the category of sets, this is exactly set comprehension, and when it is the list monad, we obtain list comprehensions familiar from programming languages.

### 2.2.2 Composing monads

Let $(\mathbf{S}, \eta^{\mathbf{S}} : 1 \to \mathbf{S}, \mu^{\mathbf{S}} : \mathbf{S}^2 \to \mathbf{S})$ and $(\mathbf{T}, \eta^{\mathbf{T}} : 1 \to \mathbf{T}, \mu^{\mathbf{T}} : \mathbf{T}^2 \to \mathbf{T})$ be two monads on $C$. It is well-known that the composition of two arbitrary monads does not automatically give rise to a new monad $\mathbf{T} \circ \mathbf{S} : C \to C$ [Beck, 1969]. One way to give a monadic structure to the composite endofunctor $\mathbf{T} \circ \mathbf{S}$ is by defining a

natural transformation $l : \mathbf{ST} \to \mathbf{TS}$, known as a distributive law [Beck, 1969]. This distributive law needs to satisfy the following axioms.




With this we can construct a monad $(\mathbf{TS}, \eta^{\mathbf{TS}}, \mu^{\mathbf{TS}})$ where

$$\eta^{\mathbf{TS}} = 1 \xrightarrow{\eta^{\mathrm{T}}\eta^{\mathrm{S}}} \mathbf{TS}$$
$$\mu^{\mathbf{TS}} = \mathbf{TSTS} \xrightarrow{\mathbf{T}l\mathbf{S}} \mathbf{TTSS} \xrightarrow{\mu^{\mathrm{T}}\mu^{\mathrm{S}}} \mathbf{TS}.$$

# Chapter 3

# Bags

A bag, also known as a multiset, is a mathematical structure that enables one to store and manipulate a collection of elements in which each element may occur more than once. Bags are similar to sets in that they both store a collection of elements, but unlike sets, they allow for multiple copies of the same element to be present in the collection. Bags can be useful in a variety of contexts, such as in computer science when representing data structures or in combinatorics when counting the number of ways that a given task can be accomplished. They are also useful in probability theory, where they can be used to represent the possible outcomes of an experiment in which each outcome may occur more than once. An important area of mathematics where bags find use is in the theory of point processes, which is the main subject of this work. We will discuss constructing and manipulating point processes in more detail in the next chapter, but first, we will develop a formal basis for combining bags with randomness.

This thesis is primarily about random bags, and more specifically, random finite bags. The sorts of random bags we would like to consider may have a random number of elements defined by some probability distribution and additionally the property that the elements themselves can be random. An example of such a random bag is the locations of earthquakes on the Earth's surface. The locations of earthquakes can be modeled as points in the two-dimensional plane, with each point representing the longitude and latitude of an earthquake. In such a dataset the number of earthquakes in any time period is a random number and their locations are distributed in a way where their occurance is most frequent near fault-lines.

Another example in a similar spirit is modelling the locations of lightning strikes in an area. To more accurately study lightning strikes as a probabilistic process, one needs to take into consideration that – contrary to popular belief – lightning can (and often does) strike the same place multiple times, and this is especially true for tall and isolated objects. This suggests the importance of multiplicities of points. In a more discrete setting, random bags arise in the study of load balancing where tasks and resources are sometimes modelled as "balls-into-bins" simulations, where multiple balls can land into the same bin. More broadly, bags, multisets and urns play a fundamental role in statistics and arise at various points in a categorical treatment [Jacobs, 2019a,b, Jacobs and Staton, 2020].

These examples motivate the study of point processes not just as random collections (sets) of points, but specifically as random bags of points where overlaps (that is, multiplicities of points) can occur. In order to study such random bags of uncountable spaces we need to work in the setting of measurable spaces, where the notion of a "sample space" of bags can be precisely formulated. To this end, in the rest of this chapter we will be working in the category of measurable spaces **Meas**.

*Unless otherwise stated, all the bags we consider in this chapter are finite bags. We examine infinite bags in a measurable setting in the final section of this chapter.*

**Outline.** We start with reviewing the measurable space of finite bags in Section 3.2 and motivate and define various constructions in detail. Our approach will be to view the measurable space of bags not as a quotient space, but as a measurable space generated by primitives that enable us to query the elements of finite bags. Following this we explain what it means to define a measure on this measurable space in Section 3.3, where we further analyse the $\sigma$-algebra for finite bags and construct a ring of sets of bags, using which we can define measures on bags by defining them on the ring instead of the entire $\sigma$-algebra. In Section 3.4 we look into defining functions on bags such that they are measurable by construction. We do this via a fold-construction but for bags, similar to the family of fold functions in functional programming languages and prove (Theorem 3.4.14) that bags can be measurably folded. In Sections 3.5 and 3.6 we study the measurable bags through an algebraic lens, presenting a monoidal (Theorem 3.5.1) and monadic structure (Theorem 3.6.6). We conclude our discussion on finite bags in Section 3.7 with showing that the measurable space of bags we have been working on so far can also be seen as the corresponding quotient measurable space (Theorem 3.7.4), tying

everything up nicely. Finally we have a look at infinite bags in Section 3.8 where we discuss some issues that appear when studying infinite bags of reals.

## 3.1 Motivating measurable spaces of bags

Let $X$ be a set. A bag is a finite unordered list of elements of $X$, or more formally an equivalence class of lists under permutation. Equivalently, a bag is a function $b : X \to \mathbb{N}$ such that $\{x \mid b(x) \neq 0\}$ is finite. For example, the bag $[\![1, 1, 0, 2, 1]\!]$ on the natural numbers can be defined as the following function $b : \mathbb{N} \to \mathbb{N}$.

$$b(n) = \begin{cases} 1, & \text{if } n = 0 \\ 3, & \text{if } n = 1 \\ 1, & \text{if } n = 2 \\ 0, & \text{otherwise} \end{cases}$$

Such a formulation extends in a natural way to defining bags as integer-valued *measures*, and this is notion of lifting bags into a measurable setting is often the preferred representation in standard point process textbooks [Daley and Vere-Jones, 2006]. Taking the powerset of $\mathbb{N}$ to be our $\sigma$-algebra, the bag above translates to the measure $b' : \Sigma_{\mathbb{N}} \to \mathbb{N}$ where now $b'(\{n\}) = b(n)$. Due to the additivity property of measures we can also query the total number of elements in the bag belonging to some given set, such as $b'(\{0, 1\}) = b'(\{0\}) + b'(\{1\}) = 4$.

For such a correspondence between bags and integer-valued measures to be possible we need to work with measurable spaces where all the points are separable. In other words these are measurable spaces where all the singleton sets are measurable. Indeed standard Borel spaces have all singletons measurable and are a standard setting for point process theory. But in more abstract settings it is possible to have $\sigma$-algebras without the presence of singletons. This is not a problem in Daley and Vere-Jones [2006] as they make the assumption that they are working on standard Borel spaces, where this correspondence holds.

As an example, consider the two-point space $\mathbf{2} = \{a, b\}$ equipped with the indiscrete $\sigma$-algebra $\Sigma_{\mathbf{2}} = \{\varnothing, \{a, b\}\}$ (to get a standard Borel space, the set $\mathbf{2}$ is equipped with the *powerset* $\sigma$-algebra). Here it is not possible to regard the singleton bag $[\![a]\!]$ as a measure because of the absence of the singleton set $\{a\}$ in the $\sigma$-algebra.

In other words, for this $\sigma$-algebra, there is no natural correspondence between bags and measures representing them.

In general, our aim is to present a theory of measurable bags for *all* measurable spaces, and not just those that are standard Borel. In the following, we will first define the measurable space of finite bags abstractly, via a universal property. Having given such a definition, we will present an equivalent characterisation of this measurable space with an explicitly constructed $\sigma$-algebra, which will aid us in directly proving measurability results.

## 3.2 The measurable space of finite bags

We begin by reviewing finite bags in **Set**. First we define the endofunctor for *bags of fixed size* $n$ in **Set**, $\mathbf{B}_n : \mathbf{Set} \to \mathbf{Set}$, as a set of $n$-tuples quotiented by the reordering relation. A bag in $\mathbf{B}_n(X)$ contains a total of $n$ elements including duplicates. We then lift this construction into the category of measurable spaces by equipping the space of bags with some corresponding $\sigma$-algebra. We use this section to spell out many of the details in this construction. A lot of this material is known to researchers working on the more measure theoretic aspects of point process theory, and it is often implicitly assumed when working with measurable spaces of bags [Macchi, 1975, Moyal, 1962].

**Definition 3.2.1.** Given a set $X$, $\mathbf{B}_n(X)$ is the set of all tuples $X^n$ quotiented by the relation $\sim_n \subseteq X^n \times X^n$ where $(x_1, \ldots, x_n) \sim_n (y_1, \ldots, y_n)$ if they are equal up to permutation.

$$\begin{aligned} &\mathbf{B}_n : \mathbf{Set} \to \mathbf{Set} \\ &\mathbf{B}_n(X) = X^n \big/ \sim_n \end{aligned}$$

The functorial action of $f : X \to Y$ is given by element-wise application.

$$\begin{aligned} &\mathbf{B}_n(f) : \mathbf{B}_n X \to \mathbf{B}_n Y \\ &\mathbf{B}_n(f)(\llbracket x_1, \ldots, x_n \rrbracket) = \llbracket f(x_1), \ldots, f(x_n) \rrbracket \end{aligned}$$

Then the finite bag endofunctor in **Set**, $\mathbf{B} : \mathbf{Set} \to \mathbf{Set}$, is simply the coproduct of

these endofunctors (which in this case is the disjoint union of sets).

$$\mathbf{B} : \mathbf{Set} \to \mathbf{Set}$$
$$\mathbf{B}(X) = \coprod_{n \in \mathbb{N}} \mathbf{B}_n(X)$$

In order to talk about the measurable space of bags we need to equip $\mathbf{B}(X)$ with some suitable $\sigma$-algebra. This $\sigma$-algebra needs to be such that it allows us to *measurably* talk about bags, and furthermore it needs to be the smallest such algebra. What sorts of queries should one be able to make about bags? The first to come to mind is to be able to ask about the number of occurrences of any $x \in X$ in a given bag $b \in \mathbf{B}(X)$. In the measurable setting $X$ is also a measurable space, i.e., it has a $\sigma$-algebra $\Sigma_X$ attached to it. Because of this, the more natural option is be able to query the number of *total occurrences* of multiple simultaneous elements in some measurable subset $W \in \Sigma_X$ of $X$ (such as the number of occurrences of $\{0, 1\}$ in the bag $[\![1, 1, 0, 2, 1]\!]$), instead of the number of occurrences of a single $x \in X$.

This can be realised as a family of functions $\#_W : \mathbf{B}(X) \to \mathbb{N}$, where the sets $W \in \Sigma_X$ are collections of elements of $X$ – let us call it an *evaluation map*. For example, if $X$ is the standard Borel space of reals and $b = [\![0.1, 1.1, 0, 0.1, 8]\!] \in \mathbf{B}(\mathbb{R})$ is the bag of interest, we have:

- $\#_{[0,1]}(b) = 3$ (elements in the unit interval),
- $\#_{\{0.1\}}(b) = 2$ (multiplicity of 0.1),
- $\#_{(8,\infty]}(b) = 0$ (elements greater than 8),
- $\#_{\varnothing}(b) = 0$ (elements in the empty set, vacuously 0), and
- $\#_{\mathbb{R}}(b) = 5$ (total number of elements).

And so, we bring this querying/evaluation ability into the measurable setting by requiring all such maps $\#_W : \mathbf{B}(X) \to \mathbb{N}$ to be measurable. As in **Set**, we will define $\mathbf{B} : \mathbf{Meas} \to \mathbf{Meas}$ via the endofunctors $\mathbf{B}_n : \mathbf{Meas} \to \mathbf{Meas}$, for which we will consider $\#^n_W : \mathbf{B}_n(X) \to \mathbb{N}$, which is the size-indexed version of $\#_W : \mathbf{B}(X) \to \mathbb{N}$. After providing this definition we will give a concrete formulation of the $\sigma$-algebra for $\mathbf{B}_n(X)$ in Lemma 3.2.6, showing it satisfies this definition.

**Definition 3.2.2.** Let $(X, \Sigma_X)$ be a measurable space, $\mathbf{B}_n(X)$ the set of bags of $X$

of size $n$, and $\mathbf{B}(X)$ the set of all bags on $X$. We define the endofunctor of bags of fixed size $n$ as

$$\begin{aligned}\mathbf{B}_n &: \mathbf{Meas} \to \mathbf{Meas}\\ \mathbf{B}_n(X, \Sigma_X) &= (\mathbf{B}_n(X), \Sigma_{\mathbf{B}_n(X)})\end{aligned}$$

where $\Sigma_{\mathbf{B}_n(X)}$ is the smallest $\sigma$-algebra such that $\#^n_W : \mathbf{B}_n(X) \to \mathbb{N}$ is measurable for all $W \in \Sigma_X$ and $n \in \mathbb{N}$ (we abuse notation and use $\mathbf{B}_n$ to refer to both, the endofunctor on **Set**, and the one on **Meas**). Finally, the endofunctor for all finite bags is given by the coproduct of all the $\mathbf{B}_n$'s.

$$\begin{aligned}\mathbf{B} &: \mathbf{Meas} \to \mathbf{Meas}\\ \mathbf{B}(X, \Sigma_X) &= \bigsqcup_{n \in \mathbb{N}} \mathbf{B}_n(X, \Sigma_X)\end{aligned}$$

We defer proving that $\mathbf{B}_n(f) : \mathbf{B}_n(X) \to \mathbf{B}_n(Y)$ is a measurable function (when $f : X \to Y$ is measurable) after providing a more explicit construction for $\Sigma_{\mathbf{B}_n}$ in Lemma 3.2.6.

**Remark 3.2.3.** Why do we not also include evaluation maps indexed by individual elements? In other words, why only consider $\#_W : \mathbf{B}(X) \to \mathbb{N}$ (where $W \in \Sigma_X$) and not also, say, $\#'_x : \mathbf{B}(X) \to \mathbb{N}$ (where $x \in X$)? Adding such evaluation maps and requiring them to be measurable will not allow us to lift a measurable space $(X, \Sigma_X)$ to a measurable bag-space in a natural way. Instead, since we are working with measurable spaces, it will require the $\sigma$-algebra of $X$ to also contain all the singleton sets. In real-world uses this is, of course, often a desirable aspect of a measurable space (for example, standard Borel spaces contain all singletons), but it prevents us from talking about *all* measurable spaces (those which may not contain singletons) in a general way.

Definition 3.2.2 is easy to state, but not so useful when proving specific properties concerning the $\sigma$-algebra $\Sigma_{\mathbf{B}_n(X)}$ as we haven't provided a direct definition for it. So far we have only described $\Sigma_{\mathbf{B}_n(X)}$ to be the smallest set so that all maps $\#^n_W$ are measurable. By definition of measurability this means $\Sigma_{\mathbf{B}_n(X)}$ needs to contain all the sets $\left(\#^n_W\right)^{-1}[U]$, where $U \in \mathcal{P}(\mathbb{N})$ is any set of naturals. Recall that the $\sigma$-algebra for $\mathbb{N}$ is generated by the singletons $\{k\}_{k \in \mathbb{N}}$. Using Lemma 2.1.7 we can rephrase this measurability requirement to obtain the condition that for all $k \in \mathbb{N}$,

$\left(\#^n_W\right)^{-1}\left[\{k\}\right] \in \Sigma_{\mathbf{B}_n(X)}$. In words, $\Sigma_{\mathbf{B}_n(X)}$ needs to contain all bags of size $n$ such that $k$ of those elements belong to $W$. An example of such a set, when $X = \mathbb{R}$, is "the set of 3-element bags such that two elements in it belong to the unit interval", which contains bags like $[\![0.1, 0.1, 3]\!]$ and $[\![0.4, 2, 0.9]\!]$ but not $[\![0, 0]\!]$ or $[\![0.1, 3, 3]\!]$. Below we define the family of sets $\beta^U_{k,n}$ to be these sets (in the example just now we were describing $\beta^{[0,1]}_{2,3}$). On its own, this family of sets is not equal to $\Sigma_{\mathbf{B}_n(X)}$. It is not even closed under intersections. To remedy this we will later generate a $\sigma$-algebra out of this family by closing it under countable unions and intersections and complements. We will then prove that the $\sigma$-algebra obtained from this process indeed satisfies the definition of $\Sigma_{\mathbf{B}_n(X)}$.

**Definition 3.2.4** (Generating sets for measurable bags)**.** Given measurable space $(X, \Sigma_X)$, define the set of bags $\beta^W_{k,n}$ to be

$$\beta^W_{k,n} = \{b \in \mathbf{B}_n(X) \mid b \text{ is an } n\text{-element bag with } k \text{ of its elements in } W\}.$$

Define $\overline{\beta}_n$ to be the set of all these $\beta$-sets.

$$\overline{\beta}_n = \left\{ \beta^U_{k,n} \;\middle|\; U \in \Sigma_X \text{ and } 0 \leq k \leq n \in \mathbb{N} \right\}$$

**Example 3.2.5.** Going back to the previous example, we have that $[\![0.1, 0.1, 3]\!]$, $[\![0.4, 2, 0.9]\!] \in \beta^{[0,1]}_{2,3}$ and $[\![0, 0]\!]$, $[\![0.1, 3, 3]\!] \notin \beta^{[0,1]}_{2,3}$. Many other sets of bags can be formed by taking appropriate intersections and union of these $\beta$-sets. For example, the set of *all* bags containing $k$ elements in $U$ can be given by the countable union of all bags of increasing size from $k$, $\bigcup_{n=k}^{\infty} \beta^U_{k,n}$. Another example is the set of all bags such that two of the four elements come from $U_1$ and the remaining two from $U_2$ (where $U_1$ and $U_2$ are disjoint). This is just the intersection $\beta^{U_1}_{2,4} \cap \beta^{U_2}_{2,4}$.

We now prove that $\overline{\beta}_n$ generates the $\sigma$-algebra $\Sigma_{\mathbf{B}_n(X)}$.

**Lemma 3.2.6.** *The endofunctor* $\mathbf{B}_n : \mathbf{Meas} \to \mathbf{Meas}$ *sends the measurable space* $(X, \Sigma_X)$ *to* $(\mathbf{B}_n(X), \Sigma_{\mathbf{B}_n(X)})$*, the space of bags of cardinality* $n$ *equipped with the* $\sigma$*-algebra* $\Sigma_{B_n(X)}$ *given by*

$$\Sigma_{B_n(X)} = \sigma(\overline{\beta}_n) = \sigma\left(\left\{ \beta^U_{k,n} \;\middle|\; U \in \Sigma_X \text{ and } 0 \leq k \leq n \in \mathbb{N} \right\}\right).$$

*(Note again the abuse of notation above. In stating* $\mathbf{B}_n(X, \Sigma_X) = (\mathbf{B}_n(X), \Sigma_{\mathbf{B}_n(X)})$ *we*

*use the endofunctor* $\mathbf{B}_n$ *on* **Meas** *on the left and the one on* **Set** *on the right.)*

*Proof.* To show that we have the required $\sigma$-algebra we need to show two things. First, that the $\sigma$-algebra we define is sufficient for all $\#^n_W : \mathbf{B}_n(X) \to \mathbb{N}$ (where $n \in \mathbb{N}$ and $W \in \Sigma_X$) to be measurable, and second that it is the minimal such $\sigma$-algebra. For $\#^n_W$ to be measurable we need the inverse image map $\left(\#^n_W\right)^{-1}[U]$ to be in our $\sigma$-algebra whenever $U$ is some subset of the naturals. Since the $\sigma$-algebra for the naturals is generated by the singleton sets, all we need to show is that for all $k \in \mathbb{N}$ (with $0 \leq k \leq n$), $\left(\#^n_W\right)^{-1}\left[\{k\}\right]$ is in the candidate $\sigma$-algebra. But this is just the set $\beta^W_{k,n}$, which was used in generating our $\sigma$-algebra, and so it is an element of it too. Finally our $\sigma$-algebra is by definition the smallest such algebra containing all the $\beta$-sets (from the definition of $\sigma$-algebra generation), which concludes the proof. □

The functorial action of $f : (X, \Sigma_X) \to (Y, \Sigma_Y)$ remains the same as in **Set**.

$$\begin{aligned}
&\mathbf{B}_n(f) : (\mathbf{B}_n X, \Sigma_{\mathbf{B}_n X}) \to (\mathbf{B}_n Y, \Sigma_{\mathbf{B}_n Y}) \\
&\mathbf{B}_n(f)(\llbracket x_1, \ldots, x_n \rrbracket) = \llbracket f(x_1), \ldots, f(x_n) \rrbracket.
\end{aligned}$$

$\mathbf{B}_n(f)$ is measurable if $\mathbf{B}_n(f)^{-1}[-]$ sends measurable sets in $\Sigma_{\mathbf{B}_n Y}$ to those in $\Sigma_{\mathbf{B}_n X}$. From Lemma 2.1.7 it suffices to check this condition on the generating sets $\beta^U_{k,n}$ where $U \in \Sigma_Y$.

$$\begin{aligned}
\mathbf{B}_n(f)^{-1}[\beta^U_{k,n}] &= \left\{ \llbracket x_1, \ldots, x_n \rrbracket \mid \mathbf{B}_n(f)(\llbracket x_1, \ldots, x_n \rrbracket) \in \beta^U_{k,n} \right\} \\
&= \left\{ \llbracket x_1, \ldots, x_n \rrbracket \mid \llbracket f(x_1), \ldots, f(x_n) \rrbracket \in \beta^U_{k,n} \right\} \\
&= \left\{ \llbracket x_1, \ldots, x_n \rrbracket \mid \llbracket x_1, \ldots, x_n \rrbracket \in \beta^{f^{-1}[U]}_{k,n} \right\} \\
&= \beta^{f^{-1}[U]}_{k,n}
\end{aligned}$$

Since $f$ is a measurable function we have that $f^{-1}[U] \in \Sigma_X$ and so $\mathbf{B}_n(f)^{-1}\left[\beta^U_{k,n}\right] = \beta^{f^{-1}[U]}_{k,n} \in \Sigma_{\mathbf{B}_n X}$. Furthermore $\mathbf{B}_n(f)$ is well-defined and respects equivalence classes.

Finally we define the space of all bags (i.e. bags of arbitrary cardinality) as the coproduct of $\mathbf{B}_n$ over all $n \in \mathbb{N}$.

**Definition 3.2.7.** The endofunctor $\mathbf{B} : \mathbf{Meas} \to \mathbf{Meas}$ sends the measurable space

$(X, \Sigma_X)$ to the sum over all $n$ of $\mathbf{B}_n(X, \Sigma_X)$.

$$\mathbf{B}(X, \Sigma_X) \equiv \left(\mathbf{B}(X), \Sigma_{\mathbf{B}(X)}\right) = \coprod_{n=0}^{\infty} \mathbf{B}_n(X, \Sigma_X) = \coprod_{n=0}^{\infty} \left(\mathbf{B}_n(X), \Sigma_{\mathbf{B}_n(X)}\right)$$

Since $\mathbf{B}_i(X)$ is disjoint from $\mathbf{B}_j(X)$ for $i \neq j$, we get the following description of the measurable space of bags $\mathbf{B}(X)$ and its $\sigma$-algebra $\Sigma_{\mathbf{B}(X)}$.

$$\mathbf{B}(X) = \coprod_{n=0}^{\infty} \mathbf{B}_n(X)$$

$$\Sigma_{\mathbf{B}(X)} = \left\{ V \subseteq \coprod_{n=0}^{\infty} \mathbf{B}_n(X) \;\middle|\; \forall n \in \mathbb{N}.\, V \cap \mathbf{B}_n(X) \in \Sigma_{\mathbf{B}_n(X)} \right\}$$

*Hereafter we omit the parentheses when writing* $\mathbf{B}(X)$.

Even though we now have an explicit description of the $\sigma$-algebra for the measurable space of bags, it is not too convenient to work with when having to prove measurability of functions with $\mathbf{B}X$ in the codomain. Take, for example, a function $f : Y \to \mathbf{B}X$. In order to prove such a function measurable, we will need to show that for each $V \in \Sigma_{\mathbf{B}X}$, $f^{-1}[V] \in \Sigma_Y$. $\Sigma_{\mathbf{B}X}$ is a large set and each $V$ is defined as the disjoint union of subsequent measurable subsets of $\mathbf{B}_nX$. Instead of this, it will be nice to show that this $\sigma$-algebra can be *generated* by some suitable family of subsets of $\mathbf{B}X$. This will simplify things by reducing the proof burden to just checking that the inverse images (under $f$) of the generating sets get sent to measurable subsets of $Y$ (Lemma 2.1.7).

Intuitively, the difficulty in considering a generating family for $\Sigma_{\mathbf{B}X}$ in terms of $\beta$-sets stems from the fact that they largely describe disjoint sets of bags (when the size indices are different). Having some sort of overlapping sets of bags will better enable us to cut them up in various ways, which is the sort of surgery that will be required for the more involved results.

To this end, rather than working with sets of bags of some fixed size containing some number of elements from a measurable set, such as $\beta^U_{k,n}$, let us consider sets of bags of arbitrary size. Below we define the family of sets $\alpha^U_k$ where $U \in \Sigma_X$ and $k \in \mathbb{N}$, which are sets of bags such that exactly $k$ of the elements of the bag belong to $U$.

**Definition 3.2.8.** Given a measurable space $(X, \Sigma_X)$, define the set $\alpha_k^U$ to be the set of bags of $X$ such that they contain $k$ elements belonging to $U$. Such a set can be defined as the infinite union of corresponding $\beta$-sets of size at least $k$.

$$\alpha_k^U = \bigcup_{n=k}^{\infty} \beta_{k,n}^U$$

Define $\overline{\alpha} \equiv \left\{ \alpha_k^U \mid U \in \Sigma_X, k \in \mathbb{N} \right\}$.

In the remainder of this section we will work towards showing this family of subsets of $\mathbf{B}X$ generating the $\sigma$-algebra for the measurable space of bags, $\Sigma_{\mathbf{B}X}$.

**Remark 3.2.9.** Note that the set $\alpha_k^U$ does not contain any bags of size smaller than $k$.

Note also that we have not lost any expressive power when working with these new sets – it is still possible to recover our $\beta$-sets from $\alpha$-sets.

**Proposition 3.2.10.** *$\alpha$- and $\beta$-sets can be expressed in terms of intersections and countable unions of each other.*

*Proof.* We have defined $\alpha$-sets in terms of $\beta$-sets. Going the other way, $\beta_{k,n}^U$ is the set of bags of size $n$ with $k$ elements in $U$. The set $\alpha_k^U$ is the set of bags of arbitrary size with $k$ elements in $U$. Next, the set $\alpha_k^X$, where $X$ is the universal set ($U \subseteq X$), is the set of bags with $k$ elements belonging to the universal set. But this is simply the set of bags of size exactly $k$ (since there are no elements not in the universal set $X$). By taking the intersections of these two sets we can recover the desired $\beta$-set, $\beta_{k,n}^U = \alpha_k^U \cap \alpha_n^X$. □

Now we show that closing all these $\alpha$-sets under complements, countable unions, and countable intersections recovers the disjoint union $\sigma$-algebra $\Sigma_{\mathbf{B}X}$. For this we first state some new definitions in order to make use of the Sierpinski-Dynkin $\pi$-$\lambda$ theorem, a useful result from measure theory which allows us to prove that a $\sigma$-algebra generated by a family of sets closed under finite intersections is a subset of a given $\sigma$-algebra (given other conditions).

**Definition 3.2.11** ($\pi$-system)**.** A $\pi$-system is a non-empty family of sets $P$ closed under finite intersections. That is, if $A, B \in P$ then $A \cap B \in P$.

**Definition 3.2.12** ($\lambda$-system)**.** A $\lambda$-system, also known as a Dynkin system, is a set $D$ of subsets of some set $X$ such that:

1. $\varnothing \in D$;
2. $D$ is closed under complements: if $A \in D$ then $\overline{A} \in D$;
3. $D$ is closed under countable unions of pairwise disjoint sets ($A_i \cap A_j = \varnothing$ for all $i \neq j$): if $A_1, A_2, \ldots$ is a sequence of pairwise disjoint sets in $D$ then $\bigcup_{i=1}^{\infty} A_i \in D$.

**Remark 3.2.13.** A $\sigma$-algebra is both a $\pi$-system and a $\lambda$-system.

**Theorem 3.2.14** (Sierpinski-Dynkin $\pi$-$\lambda$ theorem)**.** *If $P$ is a $\pi$-system and $D$ is a $\lambda$-system such that $P \subseteq D$ then the $\sigma$-algebra generated by $P$ is contained in $D$: $\sigma(P) \subseteq D$.*

With this we can show that the $\sigma$-algebra generated by the $\alpha$-sets is the disjoint union $\sigma$-algebra that we obtain for the measurable space of bags.

**Lemma 3.2.15.** $\Sigma_{\mathbf{B}X} = \sigma(\overline{\alpha}) = \sigma\left(\left\{\alpha_k^U \mid U \in \Sigma_X, k \in \mathbb{N}\right\}\right)$.

*Proof.* To prove this we will proceed in steps. First, note that the family of sets $\overline{\alpha}$ is not closed under finite intersections. Define its closure under finite intersections to be $\overline{\alpha}_\cap$, forming a $\pi$-system.

$$\overline{\alpha}_\cap = \left\{ \bigcap_{i=1}^{n} \alpha_{k_i}^{U_i} \;\middle|\; U_i \in \Sigma_X; k_i, n \in \mathbb{N} \right\}$$

$\Sigma_{\mathbf{B}X}$ is a Dynkin system (Remark 3.2.13). To show $\overline{\alpha}_\cap$ is contained in $\Sigma_{\mathbf{B}X}$, consider an arbitrary element of $\overline{\alpha}_\cap$, $\bigcap_{i=0}^{n} \alpha_{k_i}^{U_i}$. Using the definition of $\alpha$-sets (Definition 3.2.8) we get that this is the set $\bigcap_{i=0}^{n} \bigcup_{j=k_i}^{\infty} \beta_{k_i,j}^{U_i}$. Take arbitrary $m \in \mathbb{N}$, in order to calculate the intersection of this set with $\mathbf{B}_m X$.

$$\begin{aligned}
\mathbf{B}_m X \cap \bigcap_{i=0}^{n} \bigcup_{j=k_i}^{\infty} \beta_{k_i,j}^{U_i} &= \bigcap_{i=0}^{n} \bigcup_{j=k_i}^{\infty} \mathbf{B}_m X \cap \beta_{k_i,j}^{U_i} \\
&= \bigcap_{i=0}^{n} \begin{cases} \beta_{k_i,m}^{U_i} & \text{if } k_i \leq m \\ \varnothing & \text{otherwise} \end{cases}
\end{aligned}$$

$$= \begin{cases} \bigcap_{i=0}^{n} \beta_{k_i,m}^{U_i} & \text{if } \forall i = 1, \ldots, n.\ k_i \leq m \\ \varnothing & \text{otherwise} \end{cases}$$

Since this set is formed from intersections of sets from the $\overline{\beta}_n$ family, it is an element of $\Sigma_{\mathbf{B}_m X}$. And since this holds for all $m$, we can conclude that $\overline{\alpha}_\cap \subseteq \Sigma_{\mathbf{B}X}$.

And so, from the $\pi$-$\lambda$ theorem we have that the $\sigma$-algebra generated by $\overline{\alpha}_\cap$ is also contained in $\Sigma_{\mathbf{B}X}$: $\sigma(\overline{\alpha}_\cap) \subseteq \Sigma_{\mathbf{B}X}$. But now note that $\sigma(\overline{\alpha})$ and $\sigma(\overline{\alpha}_\cap)$ are the same set – in the latter we first close $\overline{\alpha}$ under all *finite* intersections and then close under complements and *countable* unions and intersections, and in the former we do that straightaway. Closing under finite intersections first makes no difference to the final result, and so $\sigma(\overline{\alpha}_\cap) = \sigma(\overline{\alpha}) \subseteq \Sigma_{\mathbf{B}X}$.

Now we prove that $\Sigma_{\mathbf{B}X} \subseteq \sigma(\overline{\alpha})$. Take $V \in \Sigma_{\mathbf{B}X}$. Then by definition we have that $\forall n \in \mathbb{N}$, $V \cap \mathbf{B}_n X \in \Sigma_{\mathbf{B}_n X}$. In other words, $V$ is a disjoint union of measurable subsets of $\mathbf{B}_n X$ ranging over all $n$. Each of these measurable subsets is some combination of complements and countable unions and intersections of $\beta$-sets since we know that $\Sigma_{\mathbf{B}_n X}$ is generated by $\overline{\beta}_n$ (Lemma 3.2.6). Furthermore we have just seen that each set $\beta_{k,n}^U$ is equal to the intersection $\alpha_k^U \cap \alpha_n^X$ (Proposition 3.2.10). By applying this result to all the $\beta$-sets in $V$, we can now express $V$ as the disjoint union of complements and countable unions and intersections of $\alpha$-sets. This implies $V \in \sigma(\overline{\alpha})$ and so $\Sigma_{\mathbf{B}X} \subseteq \sigma(\overline{\alpha})$, allowing us to conclude $\Sigma_{\mathbf{B}X} = \sigma(\overline{\alpha})$. □

Since the $\alpha$- and $\beta$-sets are interconvertible (Proposition 3.2.10), we also get as a corollary that the family of sets $\overline{\beta}$ generates the $\sigma$-algebra $\Sigma_{\mathbf{B}X}$.

**Corollary 3.2.16.** $\Sigma_{\mathbf{B}X} = \sigma(\overline{\alpha}) = \sigma(\overline{\beta})$.

We will, however, prefer to use $\overline{\alpha}$ as the generating family for $\Sigma_{\mathbf{B}X}$ as there is one fewer indexing variable to keep track of when compared with $\beta$-sets – in particular, we mean the size index. In this work we will not require taking into consideration the size of the bag in the cases where we directly prove measurability of functions with respect to $\mathbf{B}X$. It is nonetheless a useful result to have.

## 3.3 Measures on bags

Having defined the measurable space of bags $\mathbf{B}(X, \Sigma_X)$ we now discuss defining measures on them. That is, we need to be able to define functions $\nu : \Sigma_{\mathbf{B}X} \to \mathbb{R}^+_\infty$ satisfying $\sigma$-additivity. In particular, in the next chapter we will be interested in the probability measure on bags ($\nu : \Sigma_{\mathbf{B}X} \to [0, 1]$) we obtain after constructing the distributive law between the bag and Giry monads.

Defining such functions boils down to having to define them on arbitrary combinations of unions and intersections of our generating sets $\alpha^U_k$ (equivalently $\beta^U_{k,n}$). We simplify this task by making use of Carathéodory's extension theorem, an important result from measure theory, which shows that it suffices to simply define these functions on a certain *subset* of combinations of the generating sets of $\Sigma_{\mathbf{B}X}$ without needing to define them on all of $\Sigma_{\mathbf{B}X}$.

More specifically, our approach will be to consider a variation of the generating family of our $\sigma$-algebra and construct a separate set from it known as a *ring*. We can then define a *pre-measure* on the ring and use Carathéodory's extension theorem to extend this pre-measure to the full $\sigma$-algebra, giving us the resultant measure. We detail these steps below.

**Definition 3.3.1** (Ring of sets)**.** A non-empty family of sets $\mathcal{R}$ is called a ring if it contains the empty set and is closed under union and relative complement. That is, $A, B \in \mathcal{R}$ implies $A \cup B \in \mathcal{R}$ and $A \setminus B \in \mathcal{R}$.

**Definition 3.3.2** (Pre-measure)**.** Given a ring $\mathcal{R}$, a pre-measure $\nu_0 : \mathcal{R} \to [0, +\infty]$ is a function that maps subsets $U \in \mathcal{R}$ of a set $X$ to the extended real line $[0, +\infty]$, satisfying the properties below.

1. $\nu_0(\emptyset) = 0$.
2. For all countable collections of disjoint sets $U_i \in \mathcal{R}$, $\nu_0\left(\bigcup_i U_i\right) = \sum_i \nu_0(U_i)$.

That is, a pre-measure is the same as a measure but defined on a ring instead of a $\sigma$-algebra. Furthermore, recall that every probability (pre-)measure is $\sigma$-finite. In order to guarantee that the pre-measure can be extended to work consistently for the full $\sigma$-algebra we need to make use of Carathéodory's extension theorem which exactly ensures this works.

**Theorem 3.3.3** (Carathéodory's extension)**.** *Let $\mathcal{R}$ be a ring and $\nu : \mathcal{R} \to [0, +\infty]$*

*be a pre-measure. Then there exists a measure* $\tilde{\nu} : \sigma(\mathcal{R}) \to [0, +\infty]$ *such that* $\tilde{\nu}(S) = \nu(S)$ *for all* $S \in \mathcal{R}$*. This measure is unique when the pre-measure is* $\sigma$*-finite.*

We use this result in the next chapter to define a probability measure on bags by defining a pre-measure on a ring of bags. All probability measures, and therefore pre-measures, are $\sigma$-finite and so the resultant measure is indeed unique.

### 3.3.1 Defining a ring of sets of bags

So far we have shown that the $\sigma$-algebra for the measurable space of bags is generated by the $\alpha$-sets.

$$\Sigma_{\mathbf{B}X} = \sigma\left(\left\{\alpha_k^U \mid U \in \Sigma_X, k \in \mathbb{N}\right\}\right)$$

In order to use Carathéodory's extension theorem we need to be able to see this $\sigma$-algebra as being generated by a ring of sets of bags – call it $\mathcal{R}$.

Note that the family $\overline{\alpha}$ of $\alpha$-sets is *not* a ring as it is not even closed under unions. For example, take the set of bags such that there is exactly one element in $U \in \Sigma_X$ ($\alpha_1^U$) and the set of bags with exactly 3 elements in $U \in \Sigma_X$ ($\alpha_3^U$); these are disjoint sets. Then there is no single set $W \in \Sigma_X$ and $k \in \mathbb{N}$ such that we can express $\alpha_1^U \cup \alpha_3^U$ as $\alpha_k^W$. And so $\overline{\alpha}$ can *not* be regarded as our required ring $\mathcal{R}$ for invoking Carathéodory's extension theorem.

The problem here is that the sets of bags we are considering are quite rigid since they only allow a fixed number of elements of the bag to be present in a single fixed set. We need to be able to define a set of bags in a more fine-grained manner, such that we can express richer sets of bags with differing numbers of elements across various subsets of our base measurable space. Upon considering a family of such sets it will be easier to show its closure under unions and relative complements. Below is a definition of such a family of sets of bags.

**Definition 3.3.4** (Family of sets of bags)**.** Define $\mathcal{R}$ to be the following specific set of infinite unions of finite intersections of $\alpha$-sets where the family of base sets $\{U_j\}_{j=1}^n$ (where $U_j \in \Sigma_X$) is a finite partitioning of $X$ and is common across the

unions.

$$\mathcal{R} = \left\{ \bigcup_{i=1}^{\infty} \bigcap_{j=1}^{n} \alpha_{k_{i,j}}^{U_j} \;\middle|\; U_j \in \Sigma_X;\ \{U_j\}_{j=1}^{n} \text{ is a partition of } X;\ k_{i,j}, n \in \mathbb{N} \right\}$$

First some intuition on our construction of $\mathcal{R}$. We define it to be a collection of sets of bags where we describe bags by considering a family $\{U_j\}_j$ of mutually disjoint measurable sets covering the entire space $X$ (in other words, a partition of $X$). Each intersection $\bigcap_{j=1}^{n} \alpha_{k_{i,j}}^{U_j}$ describes the set of bags such that there are $k_{i,1}$ elements in $U_1$, $k_{i,2}$ elements in $U_2$, and so on. Note that we only take finite intersections because we are working with finite bags. That is, the cardinality of any given bag in the set $\bigcap_{j=1}^{n} \alpha_{k_{i,j}}^{U_j}$ is $\sum_{j=1}^{n} k_{i,j}$, and this number needs to be finite. Then by taking the union over all $i$ we are describing the set of bags which either have $k_{1,1}$ elements in $U_1$, $k_{1,2}$ elements in $U_2$, and so on, or $k_{2,1}$ elements in $U_1$, $k_{2,2}$ elements in $U_2$, and so on for all values of $i$. In other words we are describing a set of bags in a sum-of-products style. Such a formulation enables us to easily consider unions and relative complements of the sets in $\mathcal{R}$ which will help in demonstrating that it is indeed a ring of sets.

Note that we have defined the elements of $\mathcal{R}$ as infinite unions, but that doesn't prevent us from considering only finite unions. Below is an example of such a set of bags where we rewrite its definition using standard set theoretic identities to show that it is indeed an element of $\mathcal{R}$.

**Example 3.3.5.** Consider the set $(\alpha_1^A \cap \alpha_1^B) \cup (\alpha_0^A \cap \alpha_2^C)$ where $A, B, C \in \Sigma_X$ are mutually disjoint and $A \cup B \cup C = X$. To see this set as an element of $\mathcal{R}$, note that for any set $U \in \Sigma_X$ we have that

$$\bigcup_{i=0}^{\infty} \alpha_i^U = \mathbf{B}X$$

since this is just the disjoint partitioning of $\mathbf{B}X$. Then

$$\begin{aligned} \alpha_1^A \cap \alpha_1^B &= \alpha_1^A \cap \alpha_1^B \cap \mathbf{B}X \\ &= \alpha_1^A \cap \alpha_1^B \cap \bigcup_{i=0}^{\infty} \alpha_i^C \end{aligned}$$

$$= \bigcup_{i=0}^{\infty} \alpha_1^A \cap \alpha_1^B \cap \alpha_i^C$$

and similarly

$$\alpha_0^A \cap \alpha_2^C = \bigcup_{i=0}^{\infty} \alpha_0^A \cap \alpha_i^B \cap \alpha_2^C.$$

And so

$$\left(\alpha_1^A \cap \alpha_1^B\right) \cup \left(\alpha_0^A \cap \alpha_2^C\right) = \left(\bigcup_{i=0}^{\infty} \alpha_1^A \cap \alpha_1^B \cap \alpha_i^C\right) \cup \left(\bigcup_{i=0}^{\infty} \alpha_0^A \cap \alpha_i^B \cap \alpha_2^C\right) \in \mathcal{R}.$$

In general, a variety of sets of bags that one may want to consider can be shown to be elements of $\mathcal{R}$. Below we prove some results about $\mathcal{R}$ confirming this idea before showing $\mathcal{R}$ to be a ring, and finally showing that $\mathcal{R}$ and $\overline{\alpha}$ generate the same $\sigma$-algebra. First a result about describing the set $\alpha_n^W$ in terms of sets $\alpha_{m_i}^{W_i}$ where $\{W_i\}_{i=1}^k$ is some finite partitioning of $W$ and $0 \leq m_i \leq n$ for $i = 1, \ldots, n$.

**Proposition 3.3.6.** *Partition $W \in \Sigma_X$ into a finite family of measurable sets $\{W_i\}_{i=1}^k$. Then $\alpha_n^W$ can be expressed in terms of $\alpha_-^{W_i}$ sets as follows.*

$$\alpha_n^W = \bigcup_{\substack{m_1,\ldots,m_k \in \mathbb{N} \\ \text{such that} \\ m_1+\cdots+m_k=n}} \bigcap_{i=1}^{k} \alpha_{m_i}^{W_i}$$

*Note that the new expression is a* finite *union of* finite *intersections of certain sets.*

All we are doing in the above expression is refining the set of bags containing $n$ elements in $W$ into the set of bags such that there are $m_i$ elements in the respective $W_i$ subsets while requiring the total number of elements in the bag to remain $n$.

**Example 3.3.7.** To illustrate the proposition above, consider the set of bags such that each bag contains two elements belonging to the set $\{0, 1\}$, given by $\alpha_2^{\{0,1\}}$. We can partition $\{0, 1\}$ into $\{0\}$ and $\{1\}$. Then a bag $b$ is in $\alpha_2^{\{0,1\}}$ if and only if both the elements we are concerned with belong to either $\{0\}$ or $\{1\}$ or they each belong to one of the sets. In other words,

$$\alpha_2^{\{0,1\}} = \left(\alpha_0^{\{0\}} \cap \alpha_2^{\{1\}}\right) \cup \left(\alpha_1^{\{0\}} \cap \alpha_1^{\{1\}}\right) \cup \left(\alpha_2^{\{0\}} \cap \alpha_0^{\{1\}}\right).$$

We have defined $\mathcal{R}$ by taking appropriate unions and intersections of $\alpha^{U_j}_{k_{i,j}}$ sets, where $\{U_j\}_{j=1,\dots,n}$ is any finite partitioning of $X$. Two arbitrary sets of bags $\mathcal{R}$ can come from any such partitioning of $X$. When comparing these sets side-by-side, it will be much easier for our calculations if we are able to see these sets as being defined on the same partition of $X$.

To this end we prove a lemma that allows us to consider any two elements in $\mathcal{R}$, defined using two different partitionings, as coming from a finer common partition. The result amounts to the idea that considering sets of bags as countable unions of finite intersections of $\alpha$-sets gives us an expressive enough language for describing collections of bags with common normal form.

**Lemma 3.3.8.** *Let $P$ and $Q$ be two elements of $\mathcal{R}$ defined over their respective partitions $\{U_j\}_{j=1,\dots,m}$ and $\{V_j\}_{j=1,\dots,n}$.*

$$P = \bigcup_{i=1}^{\infty} \bigcap_{j=1}^{m} \alpha^{U_j}_{u_{i,j}} \qquad \text{and} \qquad Q = \bigcup_{i=1}^{\infty} \bigcap_{j=1}^{n} \alpha^{V_j}_{v_{i,j}}.$$

*Then there exists a family of sets $\{W_n\}_{n=1,\dots,k}$ and indices $u'_{i,j}$ and $v'_{i,j}$ such that*

$$P = \bigcup_{i=1}^{\infty} \bigcap_{j=1}^{k} \alpha^{W_j}_{u'_{i,j}} \qquad \text{and} \qquad Q = \bigcup_{i=1}^{\infty} \bigcap_{j=1}^{k} \alpha^{W_j}_{v'_{i,j}}.$$

*Proof.* $\{U_j\}_{j=1,\dots,m}$ and $\{V_j\}_{j=1,\dots,n}$ are both partitions of $X$. Combine them into a finer partition $\{W_j\}_{j=1,\dots,k}$ of $X$ by taking all the corresponding set intersections and differences. Now each $U_j$ and $V_j$ can be expressed as the disjoint union of some family $\{W_i\}_i$. Use Propostion 3.3.6 to express each $\alpha^{U_j}_{u_{i,j}}$ and $\alpha^{V_j}_{v_{i,j}}$ as some finite union of finite intersection of sets. Finally substitute those values back into the original presentations of $P$ and $Q$ and use standard set theoretic identities to arrive at the required normal form. □

Finally, a small result about the intersections we are considering in $\mathcal{R}$, after which we can prove that $\mathcal{R}$ forms a ring of sets.

**Proposition 3.3.9.** *Any two sets $\bigcap_{j=1}^{n} \alpha^{U_j}_{a_j}$ and $\bigcap_{j=1}^{n} \alpha^{U_j}_{b_j}$ are disjoint unless for all $j = 1, \dots, n$ we have $a_j = b_j$, in which case they are equal.*

*Proof.* When we fix a partition $\{U_j\}_{j=1}^{k}$ of $X$, the set of $k$-tuples $(c_1, \dots, c_k) \in \mathbb{N}^k$

partitions the space of bags $\mathbf{B}X$ (via the sets $\bigcap_{j=1}^{k} \alpha_{c_j}^{U_j}$). Modifying even one of the $c_j$ indices gives an entirely different set of bags. This can be stated a little more formally as follows.

$$\begin{aligned} x \in \bigcap_{j=1}^{n} \alpha_{a_j}^{U_j} \cap \bigcap_{j=1}^{n} \alpha_{a_j}^{U_j} &\iff x \in \bigcap_{j=1}^{n} \alpha_{a_j}^{U_j} \cap \alpha_{a_j}^{U_j} \\ &\iff \forall j \in \{1, \ldots, n\}.\ x \in \alpha_{a_j}^{U_j} \cap \alpha_{a_j}^{U_j} \\ &\iff \forall j \in \{1, \ldots, n\}.\ a_j = b_j \end{aligned}$$

□

**Lemma 3.3.10.** *$\mathcal{R}$ is a ring of sets.*

*Proof.* It is clear that $\varnothing \in \mathcal{R}$. Next we show that $\mathcal{R}$ is closed under pairwise unions and relative complements. Consider arbitrary $P$ and $Q \in \mathcal{R}$. From Lemma 3.3.8 we can express both $P$ and $Q$ using the same family of mutually disjoint base sets $U_j \in \Sigma_X$. This gives us $P = \bigcup_{i=1}^{\infty} \bigcap_{j=1}^{k} \alpha_{a_{i,j}}^{U_j}$ and $Q = \bigcup_{i=1}^{\infty} \bigcap_{j=1}^{k} \alpha_{b_{i,j}}^{U_j}$.

This allows us to write $P \cup Q$ as $\bigcup_{i=1}^{\infty} \bigcap_{j=1}^{k} \alpha_{c_{i,j}}^{U_j}$ where (say) $c_{2i,j} = a_{i,j}$ and $c_{2i+1,j} = b_{i,j}$. From this we can conclude that $P \cup Q \in \mathcal{R}$.

Now view $P$ as $\bigcup_{i=1}^{\infty} P_i$, where $P_i = \bigcap_{j=1}^{k} \alpha_{a_{i,j}}^{U_j}$, and similarly for $Q$. From Proposition 3.3.9 we have firstly that each of the $\{P_i\}_{i=1,\ldots,\infty}$ (and respectively for $Q$) sets are mutually disjoint. Furthermore using the same result each pair of $P_i$ and $Q_j$ sets are either disjoint or exactly equal. And so the set difference of $P \setminus Q$ will just be the union of those sets $P_i$ which are disjoint from all the $Q_j$ sets (it does not matter if the union is finite or countable), and this is also a set in $\mathcal{R}$. □

Having shown $\mathcal{R}$ to be a ring, we have by Carathéodory's extension theorem that any pre-measure defined on $\mathcal{R}$ extends to a measure defined on $\sigma(\mathcal{R})$. But note that we set out with the task of defining a ring such that it generates $\Sigma_{\mathbf{B}X} (= \sigma(\overline{\alpha}))$. To finish this we finally prove that $\sigma(\mathcal{R}) = \sigma(\overline{\alpha})$.

**Proposition 3.3.11.** *$\mathcal{R}$ and $\overline{\alpha}$ generate the same $\sigma$-algebra, $\Sigma_{\mathbf{B}X}$.*

*Proof.* First note that $\mathcal{R}$ contains $\overline{\alpha}$. This is because any $\alpha_k^U \in \overline{\alpha}$ can be expressed as $\bigcup_{i=1}^{\infty} \bigcap_{j=1}^{1} \alpha_k^U \in \mathcal{R}$. Now $\mathcal{R}$ is formed from taking certain specific countable unions and intersections of $\overline{\alpha}$. This makes no difference when generating a $\sigma$-algebra from

$\mathcal{R}$ since $\sigma$-algebras are closed under all countable unions and intersections. And so $\sigma(\overline{\alpha}) = \sigma(\mathcal{R})$. □

With this we can conclude, using Carathéodory's extension theorem, that to define a unique measure on $\mathbf{B}X$, it suffices to define a pre-measure on the sets $\bigcup_{i=1}^{\infty} \bigcap_{j=1}^{n} \alpha_{k_{i,j}}^{U_j}$ where $\{U_j\}_{j=1,\ldots,n}$ is some finite partition of $X$. We use this result in the next chapter (Section 4.3.2) to get a more concrete understanding of the action of the distributive law between the bag and Giry monads.

## 3.4 Measurable functions on bags

In this section we look at functions for creating and manipulating bags, with a focus on arguments about measurability. First we look at two functions and directly show them to be measurable. In the rest of the section we develop a general technique for defining other measurable functions on bags such that their measurability is guaranteed by construction.

### 3.4.1 Primitives for bag construction

The first function we consider is $\mathsf{singleton} : X \to \mathbf{B}X$, which given an element in $X$ returns the singleton bag $[\![x]\!]$ containing $x$. Note that hereafter $\Sigma_X$ and $\Sigma_{\mathbf{B}X}$ have been made implicit.

**Proposition 3.4.1.** $\mathsf{singleton} : X \to \mathbf{B}X$ *is measurable.*

*Proof.* We will use Lemma 2.1.7 to simplify the task of checking measurability to just check if the inverse image map of sets in $\overline{\alpha}$ under singleton is a measurable set in $\Sigma_X$. That is, we need to consider $\mathsf{singleton}^{-1}[\alpha_k^U] = \{x \in X \mid \mathsf{singleton}(x) \in \alpha_k^U\}$ for all $U \in \Sigma_X$ and $k \in \mathbb{N}$.

We have three cases here: $k = 0$, $k = 1$, and $k > 1$. When $k = 1$, we need to compute the inverse image of the set of bags where the single element is a member of $U$, which gives us $U$ itself. For $k = 0$, we compute the inverse image of the set of singleton bags where the single element is not a member of $U$, giving us $\overline{U}$. Finally, for $k > 1$ we end up with the empty set as the inverse image since singleton only

maps values to singleton bags. Overall,

$$\mathsf{singleton}^{-1}[\alpha_k^U] = \begin{cases} \overline{U} & \text{if } k = 0 \\ U & \text{if } k = 1 \\ \varnothing & \text{if } k > 1 \end{cases}$$

We have by assumption that $U \in \Sigma_X$. Furthermore, by definition of $\sigma$-algebras, we also have $\overline{U}, \varnothing \in \Sigma_X$. And so we can conclude that singleton is measurable. □

We next look at adding elements to bags. Since **Meas** has products, we can define it as the morphism $\mathsf{add} : X \times \mathbf{B}X \to \mathbf{B}X$. It sends a pair of an element and a bag to the bag with that element added.

**Proposition 3.4.2.** $\mathsf{add} : X \times \mathbf{B}X \to \mathbf{B}X$ *is measurable.*

*Proof.* Consider a measurable set $\alpha_k^U \in \Sigma_{\mathbf{B}X}$. This is the set of bags with exactly $k$ elements belonging to $U$. Then $\mathsf{add}^{-1}[\alpha_k^U]$ is the set of pairs $(x, b)$ such that $\mathsf{add}(x, b) \in \alpha_k^U$. In other words, each bag in $\alpha_k^U$ is decomposed into a set of pairs consisting of an element from the bag and the remaining bag. We consider the cases when $k = 0$ and when $k > 0$. In both these cases we demonstrate the existence of the inverse image in $\Sigma_{X \times \mathbf{B}X}$.

- $k = 0$: Here we consider the set of bags such that no element belongs to $U$. Then it is guaranteed that any element removed from the bag will be in $\overline{U}$ and the remaining bag still in $\alpha_0^U$, resulting in the inverse image being $\overline{U} \times \alpha_0^U$, which is in $\Sigma_{X \times \mathbf{B}X}$ since $\overline{U} \in \Sigma_X$ and $\alpha_0^U \in \Sigma_{\mathbf{B}X}$.
- $k > 0$: Here each bag in the set has a non-zero number of elements in $U$ and as a result we have two further cases depending on whether or not the element extracted from the bag is in $U$. If it is not in $U$, the pair consisting of the element and the remaining bag belongs to $\overline{U} \times \alpha_k^U$. If it belongs to $U$, the pair is an element of $U \times \alpha_{k-1}^U$ since there are now only $k - 1$ from the bag in $U$. And so the inverse image of $\alpha_k^U$ under add is the union of these two sets. Each of these sets is an element of $\Sigma_{X \times \mathbf{B}X}$; consequently so too is their union.

□

From these examples it can be seen that proving measurability for each newly

defined function can get tedious and complicated, especially when one needs to keep track of all the elements of the bag.

### 3.4.2 Measurable structural recursion on bags

Some of the more complicated functions on bags are $\mathsf{union} : \mathbf{B}X \times \mathbf{B}X \to \mathbf{B}X$, which returns the union of two given bags, and $\mathsf{concatenate} : \mathbf{B}(\mathbf{B}X) \to \mathbf{B}X$, which given a bag of bags returns the union of all the bags. With such functions which can drastically alter the sizes of bags we can see that the proofs of measurability will become increasingly combinatorial. To avoid this situation we will take a step back and consider the structure of the usual functions we would like to compute on these bags, and then prove all such functions to be measurable by construction.

When we take the union of two bags, we can observe it as the process of fixing one bag as a starting bag, and to it adding elements from the second bag one-by-one. The same is the case for concatenating all our bags. We start with an empty bag and then from the given bag extract the bags present and union them with our starting empty bag, which will keep accummulating elements and get filled up by the end of the process. In each of these cases our functions could be described using a process where some starting bag is fixed. Then we can extract each element from the given bag one-by-one and combine it with the starting bag using some primitive function (which was $\mathsf{add}$ in the case of $\mathsf{union}$ and $\mathsf{union}$ in the case of $\mathsf{concatenate}$), accummulating the result at each step. What is also important was that the elements from the bag can arrive in any order, and so the primitive function should also not depend on the order of elements. This is sometimes referred to as definition by *structural recursion*, and the function created from the primitive is called the *fold*. Indeed many other functions of interest can be defined in this way, which is exactly what we do in the sections and chapters to follow.

This is reminiscent of the fold construction from functional programming [Hutton, 1999]. For example, given a list of integers, it is possible to compute the sum of its elements by extracting elements starting at the head and calculating a running sum until we reach the tail. In this way the function sum can be defined as $\mathsf{fold}(\mathsf{plus}, 0)$ where $\mathsf{plus} : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ plays the role of the accumulating function and $0$ is the initial argument provided to $\mathsf{plus}$ along with the head of the list. If the list being considered is empty, the result of the fold is simply the initial argument provided, which in this case is $0$.

In general, fold (on lists, as in our example) is defined as a higher-order function with type signature $\text{fold} : (X \times Y \Rightarrow Y) \times Y \to (\text{List}X \Rightarrow Y)$. But note that the category of measurable spaces is not cartesian closed [Aumann, 1961], meaning there is no measurable evaluation function $\text{eval} : (X \Rightarrow Y) \times X \to Y$. For this reason, in our setting we will define an *uncurried* version of fold, as a *family of functions*. That is, given $f : X \times Y \to Y$, we will consider the existence of $\text{fold}_f : \text{List}X \times Y \to Y$. We will use the notation $\langle f \rangle$ for $\text{fold}_f$.

This same approach of defining functions via folding works for bags too, provided that our accumulating function is such that the order in which it receives its arguments does not matter. This way, given such a kind of accumulating function $f : X \times Y \to Y$, we get its fold $\langle f \rangle : \mathbf{B}X \times Y \to Y$. In the following we first define *symmetric functions* – functions for which the order of arguments does not matter when applied repeatedly – after which we show that each symmetric measurable function $f : X \times Y \to Y$ has a corresponding measurable fold $\langle f \rangle : \mathbf{B}X \times Y \to Y$.

**Definition 3.4.3.** A function $f : X \times Y \to Y$ is *symmetric* if

$$\forall x_1, x_2 \in X, y \in Y.\ f(x_1, f(x_2, y)) = f(x_2, f(x_1, y)).$$

In the rest of this section we define what it means to *measurably* fold a bag. To outline the construction, we will consider any given symmetric measurable $f : X \times Y \to Y$, using which we will construct measurable $f_n : X^n \times Y \to Y$. From this will be able to obtain $\widetilde{f_n} : \mathbf{B}_n X \times Y \to Y$. To show this function is measurable we will make use of the Monotone Class Theorem (stated later). An important prerequisite in being able to use this theorem is a result pertaining to images of the quotient map $q_n : X^n \to \mathbf{B}_n X$, namely that $q_n[W] \in \Sigma_{\mathbf{B}_n X}$ if and only if $W \in \Sigma_{X^n}$. Finally we can take the copairings of all the measurable $\widetilde{f_n} : \mathbf{B}_n X \times Y \to Y$ to obtain the measurable fold $\langle f \rangle : \mathbf{B}X \times Y \to Y$.

We first show that the quotient map $q_n = (x_1, \ldots, x_n) \mapsto [\![x_1, \ldots, x_n]\!] : X^n \to \mathbf{B}_n X$ is measurable.

**Proposition 3.4.4.** *The quotient map $q_n : X^n \to \mathbf{B}_n X$ is measurable.*

*Proof.* Consider $\beta^U_{k,n} \in \Sigma_{\mathbf{B}_n X}$. So as not to get lost in the details we will consider $n = 2$ and $k = 0, 1, 2$ and show that the other values of $n$ and $k$ proceed similarly.

So,

$$\begin{aligned} q_2^{-1}\left[\beta_{k,2}^U\right] &= \left\{(x_1, x_2) \mid q(x_1, x_2) \in \beta_{k,2}^U\right\} \\ &= \left\{(x_1, x_2) \mid [\![x_1, x_2]\!] \in \beta_{k,2}^U\right\} \end{aligned}$$

When $k = 0$, this is the set of pairs such that neither element is in $U$ (from the definition of $\beta_{0,2}^U$), and so $q_2^{-1}\left[\beta_{0,2}^U\right] = \{(x_1, x_2) \mid x_1, x_2 \notin U\} = \overline{U} \times \overline{U}$. In the $k = 1$ case we get $q_2^{-1}\left[\beta_{1,2}^U\right] = \{(x_1, x_2) \mid (x_1 \in U, x_2 \notin U) \text{ or } (x_1 \notin U, x_2 \in U)\} = U \times \overline{U} \cup \overline{U} \times U$. Finally by the same reasoning when $k = 2$ we get $q_2^{-1}\left[\beta_{2,2}^U\right] = U \times U$. In each of these cases we could express the inverse image in terms of unions of measurable sets in $\Sigma_{X^n}$.

The proofs for the remaining values of $k$ and $n$ follow in a straightforward manner, where the resultant image map will be a union of $\binom{k}{n}$ products where each product will contain $k$ instances of $U$ and $n - k$ instances of $\overline{U}$ in each combination. □

Now we introduce some prerequisite definitions for being able to state the monotone class theorem, which will allow us to define measurable folds.

**Definition 3.4.5** (Algebra of sets)**.** Let $X$ be some set and $C$ be some collection of subsets of $X$. $C$ is an algebra of sets (also called a "field of sets") when it contains the empty set and is closed under complements and finite unions and intersections. Equivalently $C$ is a ring of sets also containing $X$.

When $C$ is any family of subsets of $X$, we write $\mathcal{A}(C)$ for the algebra of sets generated by $C$. This is the smallest algebra of sets such that it contains $C$.

**Proposition 3.4.6.** *Let $C$ be some family of subsets of $X$. Then the $\sigma$-algebra generated by $C$ is equal to the $\sigma$-algebra generated by the algebra generated by $C$.*

$$\sigma(C) = \sigma(\mathcal{A}(C))$$

*Proof.* Since $C \subseteq \mathcal{A}(C)$ we can conclude that $\sigma(C) \subseteq \sigma(\mathcal{A}(C))$. For the other direction clearly $\mathcal{A}(C) \subseteq \sigma(C)$, and so $\sigma(\mathcal{A}(C)) \subseteq \sigma(\sigma(C))$. But $\sigma(\sigma(C)) = \sigma(C)$. Combining everything we get $\sigma(\mathcal{A}(C)) = \sigma(C)$. □

**Definition 3.4.7** (Monotone class)**.** A monotone class is a family $C$ of sets that is closed under

- countable monotone unions: if $A_1, A_2, \cdots \in C$ with $A_1 \subseteq A_2 \subseteq \ldots$ then $\bigcup_{i=0}^{\infty} A_i \in C$, and
- countable monotone intersections: if $B_1, B_2, \cdots \in C$ with $B_1 \supseteq B_2 \supseteq \ldots$ then $\bigcap_{i=0}^{\infty} B_i \in C$.

Let $C$ be any class of sets and define $M(C)$ to be the smallest monotone class containing $C$. It can be constructed by taking the intersection of all monotone classes containing $C$ or by transfinite recursion.

**Theorem 3.4.8** (Monotone class theorem)**.** *Let $C$ be an algebra of sets and let $\mathcal{M}$ be a monotone class containing $C$. Then $\mathcal{M}$ contains the $\sigma$-algebra generated by $C$. That is, $\sigma(C) \subseteq \mathcal{M}$.*

We use this theorem, the details of which can be found in a standard text on measure theory ([Fremlin, 2000, Halmos, 1976]) to establish a result about the images of $W \in \Sigma_{X^n}$ under $q_n : X^n \to \mathbf{B}_n X$. First note the following proposition about intersections of images of functions with finite fibres, i.e., functions $f : X \to Y$ such that $f^{-1}[\{y\}]$ is a finite set for $y \in Y$.

**Proposition 3.4.9.** *Let $f : X \to Y$ be a function with finite fibres. Consider the chain of decreasing sets $W_i \in \mathcal{P}(X)$ such that $W_i \supseteq W_{i+1}$. Then $f[\bigcap_i^{\infty} W_i] = \bigcap_i^{\infty} f[W_i]$.*

*Proof.* We will show the equality of the sets $f[\bigcap_i^{\infty} W_i]$ and $\bigcap_i^{\infty} f[W_i]$ by showing they are each subsets of the other. The relation $f[\bigcap_i^{\infty} W_i] \subseteq \bigcap_i^{\infty} f[W_i]$ is a standard set-theoretic identity and holds for all functions $f$ in general.

We need to show $\bigcap_i^{\infty} f[W_i] \subseteq f[\bigcap_i^{\infty} W_i]$. Consider two cases.

1. $f[\bigcap_i^{\infty} W_i] = \varnothing$: This implies $\bigcap_i^{\infty} W_i = \varnothing$. We need to show that $\bigcap_i^{\infty} f[W_i] = \varnothing$. Assume, for the sake of contradiction, that it is non-empty, and suppose there is some element $y \in \bigcap_i^{\infty} f[W_i]$. This means in each $W_i$ there exists an $x_i$ such that $f(x_i) = y$. $f$ has finite fibres, and so there are only a finite number of distinct values $x \in X$ such that $f(x) = y$, which implies our infinite sequence $x_0 \in W_0, x_1 \in W_1, \ldots$ only contains a finite set of distinct values – call them $x^1, x^2, \ldots, x^N$. Since $W_i \supseteq W_{i+1}$, and each of these sets contains one of the $N$ distinct $x^j$ values, the infinite intersection $\bigcap_i^{\infty} W_i$ will also contain at least one of the $x^j$'s, contradicting the very first implication that it is empty. And so $\bigcap_i^{\infty} f[W_i]$ is empty.

Combining this result with the standard set-theoretic identity of $f[\bigcap_i^\infty W_i] \subseteq \bigcap_i^\infty f[W_i]$, we get that $f[\bigcap_i^\infty W_i] = \varnothing \iff \bigcap_i^\infty f[W_i] = \varnothing$. (We use this in the next case to eliminate the possibility that $f[\bigcap_i^\infty W_i'] = \varnothing$ where $W_i'$ is some appropriately constructed descending sequence of sets.)

2. $f[\bigcap_i^\infty W_i] \neq \varnothing$: Let $y \in \bigcap_i^\infty f[W_i]$ (and so for all $i$, $y \in f[W_i]$). Define $Z = f^{-1}[\{y\}]$, and let $W_i' = W_i \cap Z$. Then we get the image of each of these $W_i'$ sets to be $\{y\}$:

$$f[W_i'] = f[W_i \cap Z] = f[W_i \cap f^{-1}[\{y\}]] = f[W_i] \cap \{y\} = \{y\}.$$

And so the countable intersection $\bigcap_i^\infty f[W_i']$ is also equal to $\{y\}$ (and is therefore non-empty). Note that since the $W_i'$s also form a decreasing chain of sets, we can use the same reasoning as in the previous case to deduce that $f[\bigcap_i^\infty W_i']$ is non-empty (since $\bigcap_i^\infty f[W_i']$ is non-empty). But $f[\bigcap_i^\infty W_i'] \subseteq \bigcap_i^\infty f[W_i']$ (standard set theoretic identity), and $\bigcap_i^\infty f[W_i'] = \{y\}$, as we have just shown. Consequently $f[\bigcap_i^\infty W_i'] = \{y\}$ (since it cannot be empty, which is the only other scenario), implying $y \in f[\bigcap_i^\infty W_i \cap Z]$, and so $y \in f[\bigcap_i^\infty W_i]$. From this we conclude $\bigcap_i^\infty f[W_i] \subseteq f[\bigcap_i^\infty W_i]$.

□

**Remark 3.4.10.** Note the importance of the function having *finite* fibres in the proposition we have just proved. The same result does not hold for arbitrary functions, which can be seen from the following simple example. Let $f : \mathbb{N} \to \mathbb{N}$ such that $f(i) = 0$ for all $i$. Then take the descending sequence of sets $W_0 = \mathbb{N}$, $W_1 = \mathbb{N} \setminus \{0\}$, $W_2 = \mathbb{N} \setminus \{0, 1\}$, and so on. Then $f[\bigcap_i^\infty W_i] = f[\varnothing] = \varnothing$ but $\bigcap_i^\infty f[W_i] = \{0\}$.

**Lemma 3.4.11.** *$W \in \Sigma_{X^n}$ if and only if $q_n[W] \in \Sigma_{\mathbf{B}_n X}$. In other words, $q[-]$ sends measurable sets in $\Sigma_{X^n}$ to measurable sets in $\Sigma_{\mathbf{B}_n X}$.*

*Proof.* Let $Q_n = \{W \in \Sigma_{X^n} \mid q_n[W] \in \Sigma_{\mathbf{B}_n X}\} \subseteq \Sigma_{X^n}$. We will show that $Q_n = \Sigma_{X^n}$. Define $R_0 = \{U_1 \times \cdots \times U_n \mid U_i \in \Sigma_X\}$ and recall that $\Sigma_{X^n} = \sigma(R_0)$. The family $R_0$ is closed under finite (and also countable) intersections but not complements or finite unions, and so it is not an algebra of sets. Form the family $R$ to be the closure

of $R_0$ under finite unions.

$$R = \left\{ \bigcup_{i=1}^{k} W_i \;\middle|\; W_i \in R_0 \right\}$$

Then $R$ is closed under finite unions, finite intersections, and complements, making it an algebra. It is, in fact, the smallest algebra containing $R_0$ and so $R = \mathcal{A}(R_0)$. Using Proposition 3.4.6 we have that $\Sigma_{X^n} = \sigma(R)$.

Next we show that $Q_n$ is a monotone class containing $R$. First we prove that $R$ is contained in $Q_n$. Consider an arbitrary element $\bigcup_i W_i \in R$ (where $W_i \in R_0$). Then $q_n[\bigcup_i W_i] = \bigcup_i q_n[W_i]$ since images preserve unions. So to show $\bigcup_i q_n[W_i]$ is in $\Sigma_{\mathbf{B}_n X}$ it suffices to prove each $q_n[W_i]$ is in $\Sigma_{\mathbf{B}_n X}$ since by the definition of $\sigma$-algebras the union will then also be in $\Sigma_{\mathbf{B}_n X}$.

Therefore consider the elements $U_1 \times \cdots \times U_n \in R_0$, which are our $W_i$ sets. The image $q_n[U_1 \times \cdots \times U_n] = \{[\![ x_1, \ldots, x_n ]\!] \mid x_i \in U_i\}$ is the set of all bags containing these points. We can define this set using combinations of our generating sets $\beta^U_{k,n}$ in the following way. We cut up all the $U_i$ sets such that we obtain a family of disjoint sets.

For example, when $n = 2$, we consider the sets $A = U_1 \cap U_2$, $B = U_1 \setminus U_2$, and $C = U_2 \setminus U_1$. Then the set $q_2[U_1 \times U_2]$ can be expressed as the set of all bags $b$ such that either $b$ contains both elements from each of the 3 disjoint sets, or it contains one element each from all the 3 unordered pairs of distinct sets. That is,

$$q_2[U_1 \times U_2] = \beta^A_{2,2} \cup \beta^B_{2,2} \cup \beta^C_{2,2} \cup (\beta^A_{1,2} \cap \beta^B_{1,2}) \cup (\beta^B_{1,2} \cap \beta^C_{1,2}) \cup (\beta^C_{1,2} \cap \beta^A_{1,2}).$$

This construction generalises in a straightforward manner for other values of $n$.

Now consider a monotone increasing chain of sets $W_1 \subseteq W_2 \subseteq \ldots$ ($W_i \in Q_n$, i.e., $q_n[W_i]$ is a measurable set in $\Sigma_{\mathbf{B}_n X}$). Then consider $\bigcup_i^\infty W_i$. To show that $q_n[\bigcup_i^\infty W_i]$ is a measurable set note that it is equal to $\bigcup_i^\infty q[W_i]$ since images preserve unions. But by assumption each of the $q[W_i]$ sets are measurable sets and so their union is also measurable, implying $\bigcup_i^\infty W_i \in Q_n$.

We cannot apply the same reasoning to a monotone decreasing chain of sets since images do not ordinarily preserve intersections. Note, however, that $q_n$ has finite fibres (the inverse image of a bag of size $n$ is the set of all $n$-tuples modulo permutations, which is a finite set). Consider the decreasing chain of sets $W_1 \supseteq W_2 \supseteq \ldots$ in $Q_n$.

Using Proposition 3.4.9 we have that $q[\bigcap_i^\infty W_i] = \bigcap_i^\infty q[W_i]$, and so $\bigcap_i^\infty W_i \in Q_n$. And so $Q_n$ is a monotone class containing $R$. From the monotone class theorem we can conclude that $\sigma(R) \subseteq Q_n$, i.e., $\Sigma_{X^n} \subseteq Q_n$. But from our definition of $Q_n$ we can see that $Q_n \subseteq \Sigma_{X^n}$, and so $Q_n = \Sigma_{X^n}$. □

With this we can define folding on bags. Given a measurable symmetric function $f : X \times Y \to Y$ we will define the family of measurable functions $f_n : X^n \times Y \to Y$ for all $n$. Using $f_n$ we can define measurable $\widetilde{f_n} : \mathbf{B}_n X \times Y \to Y$. From this we obtain the *fold* of $f$, denoted by $\langle f \rangle : \mathbf{B}X \times Y \to Y$, as the unique measurable function coming out of the coproduct of each of the $\widetilde{f_n}$ functions.

**Definition 3.4.12** (Fold of a function)**.** Let $f : X \times Y \to Y$ be a measurable symmetric function. Then define $\langle f \rangle : \mathbf{B}X \times Y \to Y$ to be the function which applies the accumulating function $f$ with initial value $y \in Y$ to each element of $b \in \mathbf{B}X$ one-by-one. Note that the order of selection of elements does not matter as $f$ is symmetric. We first define $f_n : X^n \times Y \to Y$ for $n$-tuples.

$$f_n : X^n \times Y \to Y$$
$$f_n((x_1, \ldots, x_n), y) = f(x_1, f(x_2, \ldots f(x_n, y) \ldots)).$$

This is clearly measurable, because it is just built from composition of measurable functions and product operations. Next we define $\widetilde{f_n}$ for bags of size $n$. When $n = 0$, $\widetilde{f_0}(\varnothing, y) = y$. For non-zero $n$,

$$\widetilde{f_n} : \mathbf{B}_n X \times Y \to Y$$
$$\widetilde{f_n}(\llbracket x_1, \ldots, x_n \rrbracket, y) = f_n((x_1, \ldots, x_n), y).$$

This is well-defined due to $f_n$ consisting of repeated applications of symmetric $f$. From this, we obtain the fold $\langle f \rangle$ as the unique function coming out of the coproduct of each of the functions $\widetilde{f_n}$ above, giving us $\langle f \rangle : \mathbf{B}X \times Y \to Y$.

**Lemma 3.4.13.** *$\widetilde{f_n} : \mathbf{B}_n X \times Y \to Y$ is measurable for symmetric measurable $f : X \times Y \to Y$.*

*Proof.* Consider some $W \in \Sigma_Y$. We need to show that $\widetilde{f_n}^{-1}[W]$ is an element of

$\Sigma_{\mathbf{B}_n X}$. By definition of the inverse image map and $\widetilde{f_n}$,

$$\begin{aligned}\widetilde{f_n}^{-1}[W] &= \{([\![x_1, \ldots, x_n]\!], y) \mid \widetilde{f_n}([\![x_1, \ldots, x_n]\!], y) \in W\} \\ &= \{([\![x_1, \ldots, x_n]\!], y) \mid f_n((x_1, \ldots, x_n), y) \in W\} \\ &= \{([\![x_1, \ldots, x_n]\!], y) \mid ((x_1, \ldots, x_n), y) \in f_n^{-1}[W]\}\end{aligned}$$

But this set is the image of $f_n^{-1}[W]$ under the function which sends the pair $((x_1, \ldots, x_n), y)$ to $(q_n(x_1, \ldots, x_n), y)$. Call this function $r_n : X^n \times Y \to \mathbf{B}_n X \times Y$. Then,

$$\widetilde{f_n}^{-1}[W] = r_n[f_n^{-1}[W]].$$

In order to show $\widetilde{f_n}^{-1}[W] \in \Sigma_{\mathbf{B}_n X}$ we consider the set $P_n = \{W \in \Sigma_{X^n \times Y} \mid r_n[W] \in \Sigma_{\mathbf{B}_n X \times Y}\}$. We show using the monotone class theorem that $P_n = \Sigma_{X^n \times Y}$, using which we can conclude that $\widetilde{f_n}^{-1}[W]$ is indeed a set in $\Sigma_{\mathbf{B}_n X \times Y}$ since $r[-]$ sends any measurable set in $\Sigma_{X^n \times Y}$ to a set in $\Sigma_{\mathbf{B}_n X \times Y}$.

First note that the generating set $\{U \times V \mid U \in \Sigma_{X^n}, V \in \Sigma_Y\}$ of $\Sigma_{X^n \times Y}$ contains the empty set and is closed under complements and finite unions and finite intersections. Next consider an arbitrary such $U \times V \in \Sigma_{X^n \times Y}$ from the generating set. We have that $U \times V$ is an element of $P_n$ since $r_n[U \times V] = q_n[U] \times V$, which is an element of $\Sigma_{\mathbf{B}_n X \times Y}$ since $q_n[U] \in \Sigma_{\mathbf{B}_n X}$ (Lemma 3.4.11) and $V \in \Sigma_Y$.

Consider a monotone increasing chain of sets $W_i \in P_n$. Following the same reasoning as in Lemma 3.4.11, the union of these sets is also an element of $P_n$ since images preserve unions. The monotone decreasing chain case also follows analogous reasoning as $r_n$ has finite fibres (since $q_n$ has finite fibres).

Having shown that $P_n$ is a monotone class containing the generating set of $\Sigma_{X^n \times Y}$ we can conclude using the monotone class theorem that $\Sigma_{X^n \times Y} \subseteq P_n$. From the definition of $P_n$ we also have that $P_n \subseteq \Sigma_{X^n \times Y}$ and so $P_n = \Sigma_{X^n \times Y}$. In other words, $r_n[-]$ sends measurable sets in $\Sigma_{X^n \times Y}$ to measurable sets in $\Sigma_{\mathbf{B}_n X \times Y}$. And so $\widetilde{f_n}^{-1}[W] \in \Sigma_{\mathbf{B}_n X \times Y}$. □

**Theorem 3.4.14.** *The fold $\langle f \rangle : \mathbf{B}X \times Y \to Y$ is measurable when $f : X \times Y \to Y$ is symmetric and measurable.*

*Proof.* The function $\langle f \rangle : \mathbf{B}X \times Y \to Y$ is a copairing of measurable functions $\widetilde{f_n} : \mathbf{B}_n X \times Y \to Y$, and so it is measurable too. □

## 3.5 The bag monoid on measurable spaces

As a first example of a fold, we can define the disjoint union function which takes a pair of bags and returns their multiplicity respecting union. With add as the accumulating symmetric function we can define the disjoint union of two bags as a measurable function by considering the fold of add where one bag provides all the new elements to be added to the other bag, which acts as the base case.

$$\uplus : \mathbf{B}X \times \mathbf{B}X \to \mathbf{B}X$$
$$\uplus(b_1, b_2) = \langle \mathsf{add} \rangle (b_1, b_2)$$

Using $\uplus$ we can show that the space of bags is a free commutative monoid.

**Theorem 3.5.1.** *For any measurable space $X$, $(\mathbf{B}X, \uplus, \varnothing)$ is a free commutative monoid in the category of measurable spaces.*

*Proof.* First note that $(\mathbf{B}X, \uplus, \varnothing)$ is a commutative monoid. Given any commutative monoid $(Y, +_Y, e_Y)$ and a map $f : X \to Y$ we can define $g = \langle \mathsf{monAcc} \rangle : \mathbf{B}X \times Y \to Y$ where monAcc is the composite

$$\mathsf{monAcc} = X \times Y \xrightarrow{f \times Y} Y \times Y \xrightarrow{+_Y} Y.$$

monAcc is a symmetric function due to $(Y, +_Y, e_Y)$ being a commutative monoid.

$$\begin{aligned} \mathsf{monAcc}(x_1, \mathsf{monAcc}(x_2, y)) &= \mathsf{monAcc}(x_1, f(x_2) +_Y y) \\ &= f(x_1) +_Y (f(x_2) +_Y y) \\ &= f(x_2) +_Y (f(x_1) +_Y y) \\ &= \mathsf{monAcc}(x_2, \mathsf{monAcc}(x_1, y)) \end{aligned}$$

From this we obtain the unique measurable commutative monoid homomorphism $f^* : \mathbf{B}X \to Y$ as $f^*(b) = g(b, e_Y)$. □

As an aside, we remark that a fold-like operation is sometimes regarded as immediate from the free (commutative) monoid property. For example, in a cartesian closed category with list objects $X^*$, the space $Y \to Y$ is a monoid (under composition), and hence any map $X \to (Y \to Y)$ induces a canonical monoid homomorphism $X^* \to (Y \to Y)$, which is a curried form of fold. However, the category of

measurable spaces is *not* cartesian closed [Aumann, 1961, Heunen et al., 2017], and so we have recorded the existence of fold as a separate fact to the free commutative monoid property.

## 3.6 The bag monad on measurable spaces

We are now ready to define the bag monad on the category of measurable spaces. Our unit and multiplication maps will be the same as those in **Set**.

**Definition 3.6.1** (Bag monad)**.** The unit $\eta^{\mathbf{B}}_X : X \to \mathbf{B}X$ of the bag monad is the measurable function singleton defined in Section 3.4.1, sending elements $x \in X$ to the singleton bag $[\![x]\!]$. The multiplication map $\mu^{\mathbf{B}}_X : \mathbf{B}(\mathbf{B}X) \to \mathbf{B}X$ is the function sending a bag of bags to the multiplicity respecting union of those bags. We define it as the fold of the accummulating symmetric function $\mathsf{dunion} = \uplus : \mathbf{B}X \times \mathbf{B}X \to \mathbf{B}X$, defined in Section 3.5, with the initial argument $\varnothing$.

$$\mu^{\mathbf{B}}_X : \mathbf{B}(\mathbf{B}X) \to \mathbf{B}X$$
$$\mu^{\mathbf{B}}_X(b) = \langle \mathsf{dunion} \rangle (b, \varnothing)$$

By defining $\mu^{\mathbf{B}}_X$ as a fold, we have by Theorem 3.4.14 that it is measurable.

It is also possible, of course, to define $\mu^{\mathbf{B}}_X$ as the set theoretic function on bags and show its measurability directly. The complication here is that in order to show measurability we need to consider sets in $\Sigma_{\mathbf{B}X}$, i.e. sets of bags. Then we need to show that these sets of bags can be viewed as the combinations of sets in $\Sigma_{\mathbf{B}(\mathbf{B}X)}$ (sets of bags of bags) by analysing the ways in which bags can be split up into bags of bags. Such a proof is quite combinatorial, a part of which we show in the next example.

**Example 3.6.2.** Let $\mathcal{P} = {\mu^{\mathbf{B}}_X}^{-1}[\alpha^U_4]$ with $\alpha^U_4 \in \Sigma_{\mathbf{B}X}$. The number 4 can be partitioned in five ways: $\{4, 3+1, 2+2, 2+1+1, 1+1+1+1\}$. $\mathcal{P}$ is the set of bags of bags such that the arbitrary union of each bag of bags contains exactly 4 elements in $U$. We start by considering elements of this set based on their cardinalities.

- There is only one collection of bags of cardinality 1 which are members of $\mathcal{P}$. These are the bags which contain a single bag which in turn contains 4 elements in $U$. Denote this collection as $\langle 4 \rangle$.

- There are three collections of bags of cardinality 2 which are members of $\mathcal{P}$. The first contains two bags which have 4 and 0 elements in $U$, the second with 3 and 1 elements in $U$, and the third with 2 and 2 elements in $U$, respectively. We write them as $\langle 4, 0\rangle$, $\langle 3, 1\rangle$, and $\langle 2, 2\rangle$.
- Cardinality 3: $\langle 4, 0, 0\rangle$, $\langle 3, 1, 0\rangle$, $\langle 2, 2, 0\rangle$, and $\langle 2, 1, 1\rangle$.
- Cardinality 4: $\langle 4, 0, 0, 0\rangle$, $\langle 3, 1, 0, 0\rangle$, $\langle 2, 2, 0, 0\rangle$, $\langle 2, 1, 1, 0\rangle$, $\langle 1, 1, 1, 1\rangle$.
- Cardinality 5: $\langle 4, 0, 0, 0, 0\rangle$, $\langle 3, 1, 0, 0, 0\rangle$, $\langle 2, 2, 0, 0, 0\rangle$, $\langle 2, 1, 1, 0, 0\rangle$, $\langle 1, 1, 1, 1, 0\rangle$. And so on.

Each collection is definable using the generating sets, and the collections of different cardinalities are mutually disjoint. For example, $\langle 3, 1, 0\rangle = \alpha_3^{\mathbf{B}X} \cap (\alpha_1^{K_3} \cap \alpha_1^{K_1} \cap \alpha_1^{K_0})$ and $\langle 2, 1, 1, 0\rangle = \alpha_4^{\mathbf{B}X} \cap (\alpha_1^{K_2} \cap \alpha_2^{K_1} \cap \alpha_1^{K_0})$ where $K_i = \alpha_i^U$. Finally, $\mathcal{P}$ is the union of all these disjoint collections.

$$\mathcal{P} = \mu_X^{B^{-1}}[\alpha_4^U] = \langle 4\rangle \cup \langle 4, 0\rangle \cup \langle 3, 1\rangle \cup \langle 2, 2\rangle \cup \langle 4, 0, 0\rangle \cup \dots$$

In general there is no closed form expression for $\mu_X^{B^{-1}}[\alpha_k^U]$ for arbitrary $k$ since it is related to the number of partitions[1] of $k$, which itself does not have a closed form. This further complicates the task of providing a suitably formalisable and easy to check proof of the measurability of $\mu_X^{\mathbf{B}}$, which is avoided when defining it as a fold.

Next we define the strength $t_{X,Y}^{\mathbf{B}} : X \times \mathbf{B}Y \to \mathbf{B}(X \times Y)$ of the bag monad. Set theoretically, this is the function $(x, [\![y_1, \dots, y_n]\!]) \mapsto [\![(x, y_1), \dots, (x, y_n)]\!]$ which can be defined as the functorial action of the function $y \mapsto (x, y)$. In order to guarantee measurability, we define an equivalent function as a fold. The accummulating symmetric function will pair each $y_i$ with $x$ and then add it to the initial argument, which will be the empty set to start with.

**Definition 3.6.3** (Bag strength)**.** Define the accummulating function strAcc

$$\begin{aligned}&\mathsf{strAcc} : Y \times (X \times \mathbf{B}(X \times Y)) \to (X \times \mathbf{B}(X \times Y))\\ &\mathsf{strAcc}(y, (x, b)) = (x, \mathsf{add}((x, y), b))\end{aligned}$$

It is clear that it is measurable and symmetric. Then the strength $t_{X,Y}^{\mathbf{B}} : X \times \mathbf{B}Y \to$

[1] OEIS, The On-Line Encyclopedia of Integer Sequences, `http://oeis.org/A000041`

$\mathbf{B}(X \times Y)$ is the second projection of the fold of strAcc.

$$t^{\mathbf{B}}_{X,Y} : X \times \mathbf{B}Y \to \mathbf{B}(X \times Y)$$
$$t^{\mathbf{B}}_{X,Y}(x, b) = (\pi_2 \circ \langle \mathsf{strAcc} \rangle)(b, (x, \varnothing))$$

In order to prove that $t^{\mathbf{B}}$ satisfies all the strength axioms we make use of the following theorem about the uniqueness of strength in well-pointed categories due to Moggi [1991].

**Theorem 3.6.4** (Uniqueness of strength)**.** *If $(T, \eta, \mu)$ is a monad over a well-pointed category $C$ (i.e. $\forall h : 1 \to A, f \circ h = g \circ h$ implies $f = g$ for any $f, g : A \to B$) with finite products then $(T, \eta, \mu, t)$ is a strong monad over $C$ if and only if $t_{A,B}$ is the unique family of morphisms such that for all points $a : 1 \to A$ and $b : 1 \to TB$*

$$t_{A,B} \circ \langle a, b \rangle = T\langle a \circ !_B, id_B \rangle \circ b$$

*where $!_B : B \to 1$ is the unique morphism from $B$ to the terminal object 1.*

**Meas** is a well-pointed category. In our case we can directly calculate to see that $\mathbf{B}(y \mapsto (x, y))(\llbracket y_1, \ldots, y_n \rrbracket) = \llbracket (x, y_1), \ldots, (x, y_n) \rrbracket = t^{\mathbf{B}}_{X,Y}(x, \llbracket y_1, \ldots, y_n \rrbracket)$.

**Lemma 3.6.5.** *$t^{\mathbf{B}}_{X,Y} : X \times \mathbf{B}Y \to \mathbf{B}(X \times Y)$ is a strength for $\mathbf{B} : \mathbf{Meas} \to \mathbf{Meas}$.*

We conclude with the main result of this chapter.

**Theorem 3.6.6.** *$(\mathbf{B} : \mathbf{Meas} \to \mathbf{Meas}, \eta^{\mathbf{B}}, \mu^{\mathbf{B}}, t^{\mathbf{B}})$ is a strong commutative monad in the category of measurable spaces.*

*Proof.* The monad and commutativity laws hold as in **Set**. $\eta^{\mathbf{B}}$ is measurable due to Proposition 3.4.1, and $\mu^{\mathbf{B}}$ and $t^{\mathbf{B}}_{X,Y}$ are both measurable as a result of their respective definitions as folds. $t^{\mathbf{B}}$ is the unique strength. □

In functional programming it is often more convenient to work with Kleisli extensions of morphisms $f : X \to \mathbf{B}Y$, $\gg\!=_{\mathbf{B}}(f) : \mathbf{B}X \to \mathbf{B}Y$. Given $\llbracket x_1, \ldots, x_n \rrbracket \in \mathbf{B}X$, we have

$$\gg\!=_{\mathbf{B}}(f)(\llbracket x_1, \ldots, x_n \rrbracket) = f(x_1) \uplus \cdots \uplus f(x_n).$$

That is, the Kleisli extension of $f$ functorially applies $f$ to each element of the input bag $b$ and returns the union of all these resultant bags.

## 3.7 The measurable space of finite bags as a quotient space

Recall that we defined the set of bags of size $n$ as the set $X^n$ quotiented by the relation $\sim_n$ which relates two $n$-tuples if they are equal up to permutation. But we defined the $\sigma$-algebra on this set as the $\sigma$-algebra generated by the sets $\beta^U_{n,k}$ (equivalently $\alpha^U_k$). There exists another way to define $\sigma$-algebras on quotient spaces, and that is known as the quotient $\sigma$-algebra. Here we will show that the $\sigma$-algebra $\Sigma_{\mathbf{B}X}$ is *the* quotient $\sigma$-algebra for the space of finite lists modulo permutation.

**Definition 3.7.1** (Quotient $\sigma$-algebra)**.** Let $(X, \Sigma_X)$ be a measurable space and let $\sim \subseteq X \times X$ be an equivalence relation on $X$. Then the quotient space $X/\sim$ can be equipped with the largest $\sigma$-algebra such that the projection map $q : X \to X/\sim$ is measurable.

$$\Sigma_{\text{quot}} \equiv \left\{ W \subseteq X/\sim \;\middle|\; q^{-1}[W] \in \Sigma_X \right\}$$

Next we define the measurable space of finite lists of $X$ as the coproduct of the product measurable spaces $(X^n, \Sigma_{X^n})$ for all $n \in \mathbb{N}$.

**Definition 3.7.2** (Measurable space of finite lists)**.** Let $(X, \Sigma_X)$ be a measurable space. Then we have the product measurable spaces $(X^n, \Sigma_{X^n})$ for all $n \in \mathbb{N}$ where $\Sigma_{X^n}$ is the usual product $\sigma$-algebra. From this we get $(X^*, \Sigma_{X^*})$, the measurable space of all finite lists, as the coproduct of all $(X^n, \Sigma_{X^n})$.

$$X^* = \bigsqcup_{n \in \mathbb{N}} X^n \qquad\qquad \Sigma_{X^*} = \left\{ W \subseteq X^* \;\middle|\; \forall n \in \mathbb{N}.\, W \cap X^n \in \Sigma_{X^n} \right\}$$

Recall that we also defined $\mathbf{B}X$ as the coproduct of $\mathbf{B}_n X$ ranging over $n \in \mathbb{N}$. From this define $q^* : X^* \to \mathbf{B}X$ as the unique measurable function formed from the combination of copairings and coprojections of $q_n : X^n \to \mathbf{B}_n X$ for all $n \in \mathbb{N}$. Finally we define the quotient $\sigma$-algebra $\Sigma'_{\mathbf{B}X}$ on $\mathbf{B}X$ as the one generated by the projection $q^*$.

**Definition 3.7.3** (Quotient $\sigma$-algebra for bags)**.** Define $\Sigma'_{\mathbf{B}X}$ to be the largest $\sigma$-algebra such that the projection map $q^* : X^* \to \mathbf{B}X$ is measurable. That is,

$$W \in \Sigma'_{\mathbf{B}X} \iff q^{*-1}[W] \in \Sigma_{X^*}.$$

We prove that the two $\sigma$-algebras $\Sigma_{\mathbf{B}X}$ (generated by $\alpha_k^U$) and $\Sigma'_{\mathbf{B}X}$ (generated by $q^* : X^* \to \mathbf{B}X$) are equal.

**Theorem 3.7.4.** *The quotient $\sigma$-algebra $\Sigma'_{\mathbf{B}X}$ is equal to $\Sigma_{\mathbf{B}X}$.*

*Proof.* Let $W$ be some measurable set in $\Sigma'_{\mathbf{B}X}$.

$$
\begin{aligned}
& W \in \Sigma'_{\mathbf{B}X} \\
& \quad \{ \text{ by definition of quotient } \sigma\text{-algebra } \} \\
\iff & q^{*-1}[W] \in \Sigma_{X^*} \\
& \quad \{ \text{ by definition of coproduct } \sigma\text{-algebra } \} \\
\iff & \forall n \in \mathbb{N}.\ q^{*-1}[W] \cap X^n \in \Sigma_{X^n} \\
& \quad \{ X^n = q^{*-1}[\mathbf{B}_n X] \} \\
\iff & \forall n \in \mathbb{N}.\ q^{*-1}[W] \cap q^{*-1}[\mathbf{B}_n X] \in \Sigma_{X^n} \\
& \quad \{ \text{ since inverse images preserve intersections } \} \\
\iff & \forall n \in \mathbb{N}.\ q^{*-1}[W \cap \mathbf{B}_n X] \in \Sigma_{X^n} \\
& \quad \{ q^{*-1}[U] = {q_n}^{-1}[U] \text{ when } U \in \Sigma_{X^n} \} \\
\iff & \forall n \in \mathbb{N}.\ {q_n}^{-1}[W \cap \mathbf{B}_n X] \in \Sigma_{X^n} \\
& \quad \{ \text{from Lemma 3.4.11} \} \\
\iff & \forall n \in \mathbb{N}.\ q_n[q_n^{-1}[W \cap \mathbf{B}_n X]] \in \Sigma_{\mathbf{B}_n X} \\
& \quad \{ q_n \text{ is surjective and so } q_n[q_n^{-1}[U]] = U \text{ for any } U \} \\
\iff & \forall n \in \mathbb{N}.\ W \cap \mathbf{B}_n X \in \Sigma_{\mathbf{B}_n X} \\
& \quad \{ \text{ by definition of coproduct } \sigma\text{-algebra } \} \\
\iff & W \in \Sigma_{\mathbf{B}X}.
\end{aligned}
$$

Note that Lemma 3.4.11 was crucial in being able to prove this result. □

### 3.7.1 The measurable space of finite bags is standard Borel

Often in probabilistic programming we work with standard Borel spaces due to the nice properties they exhibit, such as being closed under all countable products and sums and having the diagonal map $\{(x, x) \mid x \in X \times X\}$ as a measurable subset of $X \times X$. Due to Kuratowski's theorem a measurable space $(X, \Sigma_X)$ is standard Borel when it is isomorphic to either $(\mathbb{R}, \Sigma_{\mathbb{R}})$, $(\mathbb{N}, \mathcal{P}(\mathbb{N}))$, or a finite discrete measurable

space $(X, \mathcal{P}(X))$. Having shown the measurable space of bags $\mathbf{B}X$ is a quotient space, we can show that it is standard Borel when $X$ is standard Borel.

**Lemma 3.7.5.** *Let $X$ be a standard Borel space. The measurable quotient function $q^* : X^* \to \mathbf{B}X$ has a measurable section $\mathbf{B}X \to X^*$.*

*Proof.* The idea is that we can regard $\mathbf{B}X$ as a space of sorted lists in $X^*$. Any standard Borel space is either isomorphic to the reals or countable and discrete. All of these spaces have a measurable total order $(<) \subseteq X \times X$. There may or may not be a canonical choice for a particular $X$, but it doesn't matter for the sake of this proof.

We can then use this within the language of measurable functions to write a measurable sorting function $i : X^* \to X^*$ that takes a list and returns the sorted version of it. For example, if $n = 2$, let $i(x, y) = (x, y)$ if $x < y$ and otherwise $(y, x)$.

As a set-theoretic function, this sorting function $i : X^* \to X^*$ factors through the quotient map.

$$i \;=\; X^* \xrightarrow{q^*} \mathbf{B}X \xrightarrow{s} X^*$$

It remains to show that these two functions are measurable. We know that $q^*$ is measurable as it is the unique measurable morphism formed from the combination of copairings and coprojections of $q_n : X^n \to \mathbf{B}_n X$, which we know to be measurable (Proposition 3.4.4). Finally, to see that $s$ is measurable, suppose $U \in \Sigma_{X^*}$, then we must show that $s^{-1}[U] \in \Sigma_{BX}$, i.e. that $(q^*)^{-1}[s^{-1}[U]] \in \Sigma_{X^*}$. Since $s \circ q^* = i$, and $i$ is measurable, we are done. □

**Proposition 3.7.6.** *The space $(\mathbf{B}X, \Sigma_{\mathbf{B}X})$ is standard Borel when $X$ is standard Borel.*

*Proof.* If $X$ is standard Borel, then so is $X^*$, since any countable coproduct of products of standard Borel spaces is standard Borel. Using the measurable section $s : \mathbf{B}X \to X^*$ from Lemma 3.7.5 we can see $\mathbf{B}X$ to be a retract of $X^*$.

$$\mathrm{id}_{\mathbf{B}X} = \mathbf{B}X \xrightarrow{s} X^* \xrightarrow{q^*} \mathbf{B}X$$

$\mathbf{B}X$ being a retract means it is isomorphic to a Borel subset of $\mathbb{R}$. It is well-known that Borel subsets of $\mathbb{R}$ are themselves standard Borel spaces, and so we are done. □

## 3.8 Towards infinite bags

The entire chapter thus far has been about finite bags. A natural follow-up question arises – what about infinite bags? In this last section we go over the subtleties and issues that crop up when dealing with infinite bags in a measurable setting.

The measurable space of finite bags can be seen as the measurable space of finite lists quotiented by permutations , and furthermore it is standard Borel if the base space is standard Borel (Section 3.7). Standard Borel spaces are very well-behaved spaces as they are closed under countable products and countable coproducts and allow us to define an equality predicate $X \times X \to \text{Bool}$. We will show here that unfortunately this situation does not extend to the infinite bag setting.

We first define measurable spaces of infinite sequences. Let $(X, \Sigma_X)$ be a measurable space. Then the space of infinite sequences $X^{\mathbb{N}}$ is a measurable space with the infinite product $\sigma$-algebra $\Sigma_{X^{\mathbb{N}}}$

$$\Sigma_{X^{\mathbb{N}}} = \sigma\left(\left\{ \prod_{n=1}^{\infty} U_n \;\middle|\; U_n \in \Sigma_X \right\}\right)$$

Next define the equivalence relation $\sim \subseteq X^{\mathbb{N}} \times X^{\mathbb{N}}$ such that two sequences are related if they are equal up to permutation.

$$s_1 \sim s_2 \iff \exists \text{ permutation } \pi : \mathbb{N} \to \mathbb{N} \text{ such that } \pi(s_1) = s_2.$$

Finally define the space of infinite bags $\mathbf{B}_\infty X$ as the space of infinite sequences quotiented by $\sim$, giving us a quotient map $q_\infty : X^{\mathbb{N}} \to \mathbf{B}_\infty X$. Equip this space with the quotient $\sigma$-algebra to form a measurable space.

$$\mathbf{B}_\infty X = X^{\mathbb{N}} / \sim$$

$$\Sigma_{\mathbf{B}_\infty X} = \left\{ W \subseteq \mathbf{B}_\infty X \;\middle|\; q_\infty^{-1}[W] \in \Sigma_{X^{\mathbb{N}}} \right\}$$

Let us now look into whether or not $(\mathbf{B}_\infty X, \Sigma_{\mathbf{B}_\infty X})$ is standard Borel for standard Borel $(X, \Sigma_X)$. Recall that standard Borel spaces are isomorphic to one of: $(X, \mathcal{P}(X))$ where $X$ is a finite set, $(\mathbb{N}, \mathcal{P}(\mathbb{N}))$, or $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. First consider the finite case and, in order to simplify things, let $X = \text{Bool}$. Then the space of infinite bags of booleans $\mathbf{B}_\infty\text{Bool}$ contains bags with either an infinite number of

Trues and finite number of Falses, or vice-versa, or an infinite number of both – $\mathbf{B}_\infty\mathrm{Bool} \cong \{\infty\} \times \mathbb{N} \cup \mathbb{N} \times \{\infty\} \cup \{(\infty, \infty)\}$ (note that $\mathbb{N} \times \mathbb{N}$ is not present as it represents only a finite bag). This argument generalises to any finite set and it is clear that since this set is countable it is standard Borel.

In the case for naturals, we can write $\mathbf{B}_\infty\mathbb{N}$ as the set of infinite sequences of $\mathbb{N} \cup \{\infty\}$ where the sequence denotes the count of each natural number present in the bag (which can be possibly infinite), such that the total number of elements in the bag is infinite – $\mathbf{B}_\infty\mathbb{N} = \{b \in (\mathbb{N} \cup \{\infty\})^{\mathbb{N}} \mid \sum_{n \in \mathbb{N}} b(n) = \infty\}$. This can be written as the set difference $(\mathbb{N} \cup \{\infty\})^{\mathbb{N}} \setminus \bigcup_{i \in \mathbb{N}} \{b \in (\mathbb{N} \cup \{\infty\})^{\mathbb{N}} \mid \sum_{n \in \mathbb{N}} b(n) = i\}$. Each set $\{b \in (\mathbb{N} \cup \{\infty\})^{\mathbb{N}} \mid \sum_{n \in \mathbb{N}} b(n) = i\}$ is countable, and so the countable union of these sets is also countable and therefore is a Borel subset of $(\mathbb{N} \cup \{\infty\})^{\mathbb{N}}$. ($(\mathbb{N} \cup \{\infty\})^{\mathbb{N}}$ is standard Borel as it is the countable product of the standard Borel space $\mathbb{N} \cup \{\infty\}$. This way we can see $\mathbf{B}_\infty\mathbb{N}$ as a measurable subspace of $(\mathbb{N} \cup \{\infty\})^{\mathbb{N}}$. Any measurable subspace $U$ of a standard Borel space $X$ is standard Borel if and only if $U$ is a Borel subset of $X$. We have shown $\mathbf{B}_\infty\mathbb{N}$ to be a Borel subset of $(\mathbb{N} \cup \{\infty\})^{\mathbb{N}}$, and so it is a standard Borel space.

Things get a lot more interesting when we consider the space of infinite bags on $\mathbb{R}$. Unlike for the naturals, there is no immediate way to see $\mathbf{B}_\infty\mathbb{R} = \{b \in (\mathbb{N} \cup \{\infty\})^{\mathbb{R}} \mid \sum_{r \in \mathbb{R}} b(r) = \infty\}$ as a Borel subset of $(\mathbb{N} \cup \{\infty\})^{\mathbb{R}}$ since the latter is not standard Borel due to having cardinality greater than the continuum (but note that $|\mathbf{B}_\infty\mathbb{R}| = |\mathbb{R}|$). It turns out that it is not so easy to find a separable topology on the reals such that it is induced by a complete metric, making it difficult to assign a Borel $\sigma$-algebra to the space of infinite bags of reals. Here we shall examine a natural topology on the space of infinite bags of reals; we will see that it is an unusual topology where all the open sets have a non-empty intersection.

We first define what it means for a space to be standard Borel, which will require defining topological spaces, Borel spaces, and Polish spaces. Some of this was mentioned in Chapter 2 but it is worth repeating the definitions here in some more detail.

**Definition 3.8.1** (Topological space)**.** Let $X$ be a set and $\tau$ be a collection of subsets of $X$ such that it contains the emtpy set, the set $X$, and is closed under arbitrary (even uncountable) unions, and finite intersections. $\tau$ is then called a *topology* and its elements are called the *open sets*. Then $(X, \tau)$ is called a topological space. A basis is a family $B$ of open subsets of $X$ such that every open set of the topology is

equal to the union of some sub-family of $B$ and a sub-basis is a family of open sets which generates the basis upon being closed under finite intersections.

**Definition 3.8.2** (Borel space)**.** A Borel set is any set in a topological space $(X, \tau)$ that can be formed from open sets through the operations of countable union, countable intersection, and relative complement. Given $(X, \tau)$, the Borel space associated to $X$ is the pair $(X, \mathcal{B})$ where $\mathcal{B}$ is the $\sigma$-algebra of Borel sets of $X$.

**Definition 3.8.3** (Polish space)**.** A topological space that is separable and completely metrizable is called a Polish space. A topological space is separable if it contains a countable dense subset (that is, there is a set of elements $\{x_i\}_{i\in\mathbb{N}}$ of the space such that every non-empty open subset of the space contains at least one $x_i$). $(X, \tau)$ is completely metrizable if there exists a metric $d$ on $X$ such that $(X, d)$ is a complete metric space and $d$ induces the topology $\tau$.

**Definition 3.8.4** (Standard Borel space)**.** A standard Borel space is the Borel space assigned to a Polish space.

We now give an example of a natural topology on $\mathbf{B}_\infty\mathbb{R}$ which will turn out to not be metrizable. First we define the topological space of infinite sequences of reals, which we then quotient by $\sim \subseteq \mathbb{R}^{\mathbb{N}} \times \mathbb{R}^{\mathbb{N}}$.

**Definition 3.8.5** (Topological space of sequences of reals)**.** Take the space of reals with the usual Euclidean topology, i.e., the topology generated by all the open sets $(a, b) \subseteq \mathbb{R}$. Construct the infinite product $\mathbb{R}^{\mathbb{N}}$. The basis for this topology is given by the family $B$ below.

$$B = \left\{ \prod_{n\in\mathbb{N}} U_n \;\middle|\; U_n \text{ open in } \mathbb{R};\ U_n = \mathbb{R} \text{ for all but finitely many } n \in \mathbb{N} \right\}$$

$B$ is a basis for the topology for sequences of reals $\tau_{\mathbb{R}^{\mathbb{N}}}$.

In other words, a set in this basis is a set of sequences of reals whose coordinates are unrestricted in all but finitely many indices. Examples of such sets are $\mathbb{R} \times (0, 1) \times \mathbb{R}^2 \times (3, 6) \times \mathbb{R}^{\mathbb{N}}$ and $\mathbb{R}^{100} \times (-1, 0) \cup (0, 1) \times \mathbb{R}^{\mathbb{N}}$.

**Definition 3.8.6** (Topological space of infinite bags)**.** Define the topological space $(\mathbf{B}_\infty\mathbb{R}, \tau_\infty)$ to be the quotient topological space of $\mathbb{R}^{\mathbb{N}}$ quotiented by $\sim$, giving us

the quotient map $q_\infty : \mathbb{R}^{\mathbb{N}} \to \mathbb{R}^{\mathbb{N}}/\sim$. Then we have the following definition of $\tau_\infty$.

$$\tau_\infty = \left\{ U \;\middle|\; q_\infty^{-1}[U] \in \tau_{\mathbb{R}^{\mathbb{N}}} \right\}$$

Combining all of our definitions together, we obtain the following chain of equivalences.

$$\begin{aligned}
&U \in \tau_\infty \\
&\qquad \{ \text{ by definition of } \tau_\infty \} \\
&\iff q_\infty^{-1}[U] \in \tau_{\mathbb{R}^{\mathbb{N}}} \\
&\qquad \{ \text{ by definition of the basis } B \text{ for } \tau_{\mathbb{R}^{\mathbb{N}}} \} \\
&\iff q_\infty^{-1}[U] = \bigcup_{i \in \mathcal{I}} W_i \quad (\text{where } W_i \in B) \\
&\qquad \{ \text{ by definition of the sets in } B \} \\
&\iff q_\infty^{-1}[U] = \bigcup_{i \in \mathcal{I}} \prod_{j \in \mathbb{N}} Z_j^i \quad (\text{for fixed } i,\ Z_j^i = \mathbb{R} \text{ for all but finitely many } j \in \mathbb{N}).
\end{aligned}$$

**Example 3.8.7.** An example of an open set in this quotient topology is the set $W_1$ of infinite bags such that "there is at least one element in the bag in the interval $(0, 1)$". This is because $q_\infty^{-1}[W_1] = \bigcup_{i \in \mathbb{N}} \mathbb{R}^i \times (0, 1) \times \mathbb{R}^{\mathbb{N}}$, and this is open in $\mathbb{R}^{\mathbb{N}}$ as it is the union of basis sets in $B$. Another similar example is the set $W_2$ of infinite bags such that "there is at least one element from the interval $(0, 1)$ and one from $(2, 3)$", since

$$\begin{aligned}
q_\infty^{-1}[W_2] = &\bigcup_{i,j \in \mathbb{N}} \mathbb{R}^i \times (0,1) \times \mathbb{R}^j \times (2,3) \times \mathbb{R}^{\mathbb{N}} \\
&\cup \bigcup_{i,j \in \mathbb{N}} \mathbb{R}^i \times (2,3) \times \mathbb{R}^j \times (0,1) \times \mathbb{R}^{\mathbb{N}}
\end{aligned}$$

which is open in $\mathbb{R}^{\mathbb{N}}$.

In order to determine if $\mathbf{B}_\infty\mathbb{R}$ is separable, we need to find a countable dense subset of $\mathbf{B}_\infty\mathbb{R}$. That is, we need to find a countable set such that it intersects every open set $U \in \tau_\infty$. It just so happens that we can find not only a countable set, but a *singleton* set satisfying this property.

**Lemma 3.8.8.** *Let $U$ be any open subset of $\mathbf{B}_\infty\mathbb{R}$ and let $b_{\mathbb{Q}}$ be the infinite bag of*

*reals containing all the rationals with multiplicity 1. Then* $b_{\mathbb{Q}} \in U$.

*Proof.* We have shown that $U \in \tau_\infty \iff q_\infty^{-1}[U] = \bigcup_{i \in I} \prod_{j \in \mathbb{N}} Z_j^i$ where for fixed $i$, $Z_j^i = \mathbb{R}$ for all but finitely many $j \in \mathbb{N}$. Consider any product in this arbitrary union, $\prod_{j \in \mathbb{N}} Z_j^i$. Since only a finite number of sets from this infinite product are restricted to some open proper subsets of $\mathbb{R}$, we can always find a permutation of the set of rationals, viewed as an infinite sequence, such that this sequence is an element of $\prod_{j \in \mathbb{N}} Z_j^i$ – we just need to ensure that appropriate rationals are placed in the indices with the restricted $Z_j^i$ sets. That is, we can always regard $b_{\mathbb{Q}} \in \mathbf{B}_\infty\mathbb{R}$ as some sequence $s_{\mathbb{Q}} \in \mathbb{R}^{\mathbb{N}}$ such that $s_{\mathbb{Q}} \in \prod_{j \in \mathbb{N}} Z_j^i$. The set $\bigcup_{i \in I} \prod_{j \in \mathbb{N}} Z_j^i$ is closed under all permutations due to being defined as the inverse image under $q_\infty : \mathbb{R}^{\mathbb{N}} \to \mathbf{B}_\infty\mathbb{R}$. And so $\{\pi(s_{\mathbb{Q}}) \mid \pi : \mathbb{N} \to \mathbb{N} \text{ is a permutation}\} \subseteq \bigcup_{i \in I} \prod_{j \in \mathbb{N}} Z_j^i$. But $\{\pi(s_{\mathbb{Q}}) \mid \pi : \mathbb{N} \to \mathbb{N}\} = q_\infty^{-1}[\{b_{\mathbb{Q}}\}]$ and so $\{b_{\mathbb{Q}}\} \subseteq U$, proving $b_{\mathbb{Q}} \in U$. □

In general there are continuum-many infinite bags that are common to all the open sets of $\tau_\infty$. This is the set of all infinite bags of all rationals. Just now we have shown the case for when all the multiplicities are equal to 1, but the proof remains largely the same for arbitrary multiplicities. Topological spaces where no two non-empty open sets are disjoint are known as *hyperconnected spaces*. Our space satisfies the property of being hyperconnected, but also satisfies the stronger property that the intersection of all the open sets is non-empty.

**Lemma 3.8.9.** *The topological space of infinite bags on reals is a hyperconnected space.*

And so, we get that $\mathbf{B}_\infty\mathbb{R}$ is a separable space where the countable dense set is just the singleton set of the set of rationals regarded as an infinite bag with all the elements having multiplicity 1. Because of this it is not a Hausdorff space, which implies it cannot be metrizable, stopping our attempt of proving it is Polish.

Further work needs to be done to determine whether $\mathbf{B}_\infty\mathbb{R}$ can be equipped with a $\sigma$-algebra making it into a standard Borel space. Another way to define a measurable space of infinite bags of reals would be by defining some appropriate notion of "evaluating" (or querying) an infinite bag, presenting this action as a morphism coming out of bags, and then requiring all such morphisms to be measurable. It is unclear if this would be equal to a corresponding quotient $\sigma$-algebra, however. If we proved it like we did in the case for finite bags, we would have to have a result similar to that of Proposition 3.4.9 so that we can prove $q_\infty[\bigcap_i^\infty W_i] = \bigcap_i^\infty q_\infty[W_i]$,

where $\{W_i\}_{i\in\mathbb{N}}$ is any chain of decreasing sets. But because $q_\infty$ has infinite fibres it is easy to find a counterexample to this statement. This is, of course, not a proof of inequality of the two measurable spaces, but it may indicate the problem is more complex.

It would not be too surprising if we were to find out the measurable space of infinite bags is not standard Borel. In the field of descriptive set theory, and in particular in the study of Borel equivalence relations, similar questions have been looked into before [Kechris, 1995]. A well-known example of a quotient of a standard Borel space not being standard Borel is the following: consider the Cantor space, i.e. the space $\mathbf{2}^{\mathbb{N}}$, and say two sequences are related if they only differ at a finite number of indices. Then the quotient space formed from this equivalence relation is not standard Borel [Gao, 2008].

# Chapter 4

# The point process monad

In the previous chapter we laid the foundations for the bag endofunctor in the category of measurable spaces and gave a construction for measurable structural recursion on these bags via a folding operator. Using this we were able to show how the monoidal and monad structure on bags lifts to the measurable setting. In this chapter we continue along these lines and work with the composition of the Giry monad (the monad of probability measures) and the bag monad to describe point processes.

**Outline.** We begin with a review of the Giry monad $\mathbf{G}$ in Section 4.1 and compose it with the bag monad to give a definition of the measurable space of point processes in Section 4.2. Using our fold construction from the previous chapter we define a natural transformation $\mathbf{BG} \to \mathbf{GB}$ in Section 4.3.1 and prove that it is a distributive law (Theorem 4.3.4). Later we go on to prove that the point process endofunctor is a strong monad in Theorem 4.3.8 and that it is not commutative (Theorem 4.3.9). In Section 4.3.4 we examine the Kleisli extension (monadic bind) of the point process monad. Following this we give examples of several constructions on point processes using our monad in Section 4.4. Finally we look at infinite point processes as random streams in Section 4.5. We first show that the stream endofunctor $\mathbf{S}$ on measurable spaces is a strong monad (Theorem 4.5.3). Next we prove that there is in fact no distributive law $\mathbf{SG} \to \mathbf{GS}$ in Theorem 4.5.4. We follow this with a construction of a morphism $\mathbf{SG} \to \mathbf{GS}$ in Section 4.5.3 and show that it is a weak distributive law (Theorem 4.5.9).

## 4.1 The Giry monad

Let $(X, \Sigma_X)$ be a measurable space. Then a probability measure on $X$ is a measure $\nu : \Sigma_X \to [0,1]$ (with $\nu(X) = 1$). We can collect all of these measures into a space of their own – call this set $\mathrm{G}X = \{\nu : \Sigma_X \to [0,1] \mid \nu \text{ is a probability measure on } X\}$. This space can be assigned a $\sigma$-algebra in a very natural way, making the space of probability measures a *measurable space* of probability measures. This makes $\mathrm{G} : \mathbf{Meas} \to \mathbf{Meas}$ an endofunctor in the category of measurable spaces. The key insight of Michele Giry [1982] was that this endofunctor is actually monadic. In this section we spell out the construction of the Giry monad and describe how it allows us to compose probability measures in a meaningful way. For this we will make use of monad comprehension notation to show how the act of sampling from a probability measure can be clearly understood monadically.

We have the measurable space $(X, \Sigma_X)$ and the set of all probability measures $\mathrm{G}X$ on it. We can describe the $\sigma$-algebra corresponding to $\mathrm{G}X$ in a way similar to the construction of our finite bag endofunctor $\mathrm{B} : \mathbf{Meas} \to \mathbf{Meas}$, by looking at the most natural family of functions operating on these probability measures, and then defining $\Sigma_{\mathrm{G}X}$ to be the smallest $\sigma$-algebra such that these functions are measurable. And so, just like with bags, we can *evaluate* our probability measures, and this is done by applying these measures to measurable subsets of $X$. Call this evaluation map $\mathrm{ev}_U : \mathrm{G}X \to [0,1]$, where $U \in \Sigma_X$. Then $\mathrm{ev}_U(\nu) = \nu(U) \in [0,1]$. This motivates defining $\Sigma_{\mathrm{G}X}$ to be the smallest $\sigma$-algebra such that all $\mathrm{ev}_U : \mathrm{G}X \to [0,1]$ functions are measurable. Further analysing these measurable evaluation maps allows us to calculate an explicit family of subsets which generate $\Sigma_{\mathrm{G}X}$ (exactly like in the previous chapter, where we gave the measurable space of bags the same treatment). This is the family $D_r^U$ of subsets of $\mathrm{G}X$, where $U \in \Sigma_X$ and $r \in [0,1]$, containing probability measures $\nu : \Sigma_X \to [0,1]$ such that $\nu(U) \leq r$. Then we obtain $\Sigma_{\mathrm{G}X} = \sigma(\{D_r^U \mid U \in \Sigma_X, r \in [0,1]\})$.

**Definition 4.1.1** (Giry endofunctor)**.** Let $(X, \Sigma_X)$ be a measurable space. Then $\mathrm{G}(X, \Sigma_X)$ is the measurable space of all probability measures $\nu : \Sigma_X \to [0,1]$ on $X$ endowed with the smallest $\sigma$-algebra such that the evaluation maps $\mathrm{ev}_U = \nu \mapsto \nu(U) : \mathrm{G}X \to [0,1]$ are measurable. Given measurable $f : X \to Y$, the functorial action of $\mathrm{G}(f) : \mathrm{G}X \to \mathrm{G}Y$ on $\nu \in \mathrm{G}X$ is given by the pushforward measure $\nu \circ f^{-1} \in \mathrm{G}Y$.

**Definition 4.1.2** (Giry monad)**.** The unit $\eta_X^{\mathrm{G}} : X \to \mathrm{G}X$ sends $x \in X$ to the Dirac

delta $\delta_x \in \mathbf{G}X$.

$$\delta_x = W \in \Sigma_X \mapsto \begin{cases} 1, & \text{if } x \in W \\ 0, & \text{otherwise} \end{cases}$$

The monad multiplication $\mu^{\mathbf{G}}_X : \mathbf{GG}X \to \mathbf{G}X$ sends measures of measures to a probability measure on $X$ "averaging out" all the probabilities, which we obtain by integration.

$$\mu^{\mathbf{G}}_x : \mathbf{GG}X \to \mathbf{G}X$$
$$\mu^{\mathbf{G}}_x(\nu) = U \in \Sigma_X \mapsto \int_{p \in \mathbf{G}X} \mathrm{ev}_U(p)\nu(\mathrm{d}p)$$

Furthermore the Giry monad is strong, meaning there is a map $t^{\mathbf{G}}_{X,Y} : X \times \mathbf{G}Y \to \mathbf{G}(X \times Y)$ which is measurable.

In this work we will prefer to use the monadic bind (with monad comprehension syntactic sugar) of the Giry monad, $\gg\!=_{\mathbf{G}}\ (f) = \mu^{\mathbf{G}}_Y \circ \mathbf{G}(f) : \mathbf{G}X \to \mathbf{G}Y$, where $f : X \to \mathbf{G}Y$ is some measurable function. This is due to how the operational semantics of the bind closely mirrors common sampling operations. We explain with an example. Let $m$ and $v$ be real random variables, distributed uniformly on the closed intervals $[0, 10]$ and $[1, 2]$ respectively. Given these random variables we can consider the normal distribution centered around $m$ with variance $v$, from which we can sample new values. This describes some random variable, call it $Y$, which we can express in mathematical notation as follows.

$$m \sim \mathsf{Uniform}(0, 10)$$
$$v \sim \mathsf{Uniform}(1, 2)$$
$$Y \sim \mathsf{Normal}(m, v)$$

The benefit of the monadic bind of the Giry monad is that using monad comprehensions we can define $Y$ in a very similar fashion. In particular, if $\mathsf{unif} : \mathbb{R}^2 \to \mathbf{G}\mathbb{R}$ and $\mathsf{normal} : \mathbb{R}^2 \to \mathbf{G}\mathbb{R}$ are the uniform and normal distributions respectively, then we arrive at the following definition for $Y \in \mathbf{G}\mathbb{R}$.

$$Y = \{\!|\, y \mid m \leftarrow \mathsf{unif}(0, 10), v \leftarrow \mathsf{unif}(1, 2), y \leftarrow \mathsf{normal}(m, v) \,|\!\}_G$$

We make use of this "sampling semantics" of the Kleisli category of the Giry monad

later in this chapter in order to get a better understanding of the bind of the point process monad $\mathbf{G} \circ \mathbf{B}$.

Finally, the Giry monad also restricts to standard Borel spaces – if $(X, \Sigma_X)$ is standard Borel then so is $\mathbf{G}(X, \Sigma_X)$.

## 4.2 The point process endofunctor

Recall that point processes are random collections of points, where the idea that overlapping points can exist is crucial to their definition. To this end they are sometimes defined to be random integer-valued measures (such as in [Daley and Vere-Jones, 2006]), which makes for convenient calculations, or as random tuples quotiented by a reordering relation, i.e. random bags (such as in [Macchi, 1975, Moyal, 1962]). In the previous chapter we developed machinery for defining measurable functions on bags by structural recursion, which gives us the best of both approaches: convenience and generality.

Having established the definitions and properties of the bag and Giry endofunctors $\mathbf{B}, \mathbf{G} : \mathbf{Meas} \to \mathbf{Meas}$, we now have everything we need in order to define the measurable space of point processes. Formally, $\mathbf{G}$ and $\mathbf{B}$ are endofunctors on **Meas**, and so given a measurable space $(X, \Sigma_X)$, the application of $\mathbf{G} \circ \mathbf{B}$ to it gives us the *measurable space* of point processes on $X$.

$$\mathbf{G}(\mathbf{B}(X, \Sigma_X)) \equiv \text{Measurable space of point processes on } X.$$

*From here on we only work in the category of measurable spaces and omit the $\sigma$-algebra when it is clear from context.*

Again, since $\mathbf{G}$ and $\mathbf{B}$ are endofunctors on **Meas**, we have by their composition a $\sigma$-algebra associated to the space of point processes. This can be obtained from the definition of the $\sigma$-algebra for the $\mathbf{G}X$ and $\mathbf{B}X$.

$$\Sigma_{\mathbf{GB}X} = \sigma(\{D_r^U \mid U \in \Sigma_{\mathbf{B}X}, r \in [0, 1]\})$$

In order to compose together point processes to form larger ones, we need to be able to define measurable maps into and out of $\mathbf{GB}X$. For these maps to be measurable we will need to ensure the inverse images of these maps respect the structure of $\Sigma_{\mathbf{GB}X}$, and the amount of additional work that this brings us depends on whether or

not $\mathbf{GB}X$ is in the domain or codomain of the map. The process of checking these measurability requirements for each new function we define can very easily get tedious and difficult, and this is because of the complicated structures of arbitrary $\sigma$-algebras. In most cases $\sigma$-algebras are defined via a transfinite process, and so proving measurability of maps on these $\sigma$-algebras will amount to requiring a transfinite induction argument for each new map we define.

To overcome this problem, we will consider a more abstract approach, as mentioned in the introductory chapter. In particular, we will look into exhibiting a monad on the point process endofunctor. By doing so, we will obtain two functions, the unit $\eta^{\mathrm{GB}} : 1 \to \mathbf{GB}$ and the multiplication $\mu^{\mathrm{GB}} : \mathbf{GBGB} \to \mathbf{GB}$, which will give us sufficient machinery to develop a rich family of well-behaved algebraic functions on point processes. This is the main subject of this chapter, which we begin from the next section.

## 4.3 A monad for finite point processes

With the measurable space of point processes defined using the composition of $\mathbf{G}$ and $\mathbf{B}$, we can now study the monadic structure on $\mathbf{G} \circ \mathbf{B}$. This is, unfortunately, not immediate from the fact that $\mathbf{G}$ and $\mathbf{B}$ are monads. As mentioned in Section 2.2.2, it is well-known that the composition of two monads does not automatically give rise to a new monad [Beck, 1969]. There are two solutions to this problem. We can either define a monad on $\mathbf{GB}$ from scratch by constructing appropriate unit and multiplication maps, or define a distributive law between the two monads, from which we can readily obtain a monad on $\mathbf{GB}$. Note that it is also possible that there may be no monadic structure at all on the composition of two monads. Thankfully for us, that is not the case. In the following we take the latter approach to exhibiting a monad on $\mathbf{GB}$, which is by a distributive law $l : \mathbf{BG} \to \mathbf{GB}$. We first construct a measurable family of maps $l_X : \mathbf{BG}X \to \mathbf{GB}X$, and then show that it satisfies the axioms of a distributive law.

### 4.3.1 A distributive law between the bag and Giry monads

In order to give a distributive law, we need to exhibit a morphism $l_X : \mathbf{BG}X \to \mathbf{GB}X$ from bags of probability measures to probability measures on bags. After considering the axioms that such a morphism will need to satisfy in order to be a distributive law between $\mathbf{B}$ and $\mathbf{G}$, we can come to a conclusion about how it should

operate on an element $b \in \mathbf{BG}X$. We arrive at the following idea – given a bag of probability measures $b = [\![ \nu_1, \ldots, \nu_n ]\!]$, the resultant probability measure of bags $l_X([\![ \nu_1, \ldots, \nu_n ]\!]) \in \mathbf{GB}X$ will need to be such that a sample bag from it contains only elements which are in turn sampled from each of the measures $\nu_i \in \mathbf{G}X$ in $b$. In other words, if $[\![ x_1, \ldots, x_n ]\!]$ is a sample from the resultant probability measure, we require each of these $x_i$ elements to be a sample from one of the $\nu_{i'}$ measures (where $i'$ is the result of applying some permutation to $i$). Such a map will satisfy the axioms of a distributive law, which we formally prove later. In this section we define exactly this measurable morphism using our fold construction from Chapter 3.

There are a couple of steps to defining such a morphism. First, we construct an appropriate accummulating function distrAcc which takes as input a probability measure and a probability measure of bags and returns a probability measure of bags where the value sampled from the first input is added to the bag sampled in the second input. Next we show that it is symmetric, i.e. that $\mathsf{distrAcc}(p_1, \mathsf{distrAcc}(p_2, \nu)) = \mathsf{distrAcc}(p_2, \mathsf{distrAcc}(p_1, \nu))$. Then we fold over distrAcc and fix the initial value of the accummulator to define the distributive law $l : \mathbf{BG} \to \mathbf{GB}$. Finally we show that this morphism satisfies all the requirements of a distributive law.

**Definition 4.3.1.** The accummulator $\mathsf{distrAcc}_X : \mathbf{G}X \times \mathbf{GB}X \to \mathbf{GB}X$ for the distributive law is given by the following composition.

$$\mathsf{distrAcc}_X = \mathbf{G}X \times \mathbf{GB}X \xrightarrow{\text{double strength}} \mathbf{G}(X \times \mathbf{B}X) \xrightarrow{\mathbf{G}(\mathsf{add})} \mathbf{GB}X$$

where $\mathsf{add} : X \times \mathbf{B}X \to \mathbf{B}X$ is the measurable function for adding elements to bags defined in Section 3.4.1, and the double strength is a map similar to the strength $t^{\mathbf{G}}_{X,Y} : X \times \mathbf{G}Y \to \mathbf{G}(X \times Y)$. We write it in monad comprehension notation as

$$\begin{aligned} &\mathsf{dst}_{X,Y} : \mathbf{G}X \times \mathbf{G}Y \to \mathbf{G}(X \times Y) \\ &\mathsf{dst}_{X,Y}(\nu_1, \nu_2) = \{\!\!\{ (x_1, x_2) \mid x_1 \leftarrow \nu_1, x_2 \leftarrow \nu_2 \}\!\!\}_{\mathbf{G}}. \end{aligned}$$

$\mathsf{dst}_{X,Y}(\nu_1, \nu_2)$ is the product measure $\nu_1 \otimes \nu_2$.

The accummulator $\mathsf{distrAcc}_X$ is therefore a composition of measurable functions and so it is measurable. In the proofs to follow it is more convenient to work with

distrAcc in its monad comprehension form.

$$\begin{aligned}
&\mathsf{distrAcc}_X : \mathbf{G}X \times \mathbf{GB}X \to \mathbf{GB}X \\
&\mathsf{distrAcc}_X(p, \nu) = \mathbf{G}(\mathsf{add})\{\!|\, (x, b) \mid x \leftarrow p, b \leftarrow \nu \,|\!\}_{\mathrm{G}} \\
&\qquad\qquad = \{\!|\, \mathsf{add}(x, b) \mid x \leftarrow p, b \leftarrow \nu \,|\!\}_{\mathrm{G}}
\end{aligned}$$

**Lemma 4.3.2.** $\mathsf{distrAcc}_X : \mathbf{G}X \times \mathbf{GB}X \to \mathbf{GB}X$ *is a symmetric function.*

*Proof.* We prove this by equational reasoning.

$$\begin{aligned}
&\mathsf{distrAcc}_X(p_1, \mathsf{distrAcc}_X(p_2, \nu)) \\
&\qquad \{ \text{ by definition } \} \\
&= \{\!|\, \mathsf{add}(x_1, b') \mid x_1 \leftarrow p_1, b' \leftarrow \{\!|\, \mathsf{add}(x_2, b) \mid x_2 \leftarrow p_2, b \leftarrow \nu \,|\!\}_{\mathrm{G}} \,|\!\}_{\mathrm{G}} \\
&\qquad \{ \text{ monad multiplication law } \} \\
&= \{\!|\, \mathsf{add}(x_1, \mathsf{add}(x_2, b)) \mid x_1 \leftarrow p_1, x_2 \leftarrow p_2, b \leftarrow \nu \,|\!\}_{\mathrm{G}} \\
&\qquad \{ \text{ add is symmetric and } \mathbf{G} \text{ is a commutative monad } \} \\
&= \{\!|\, \mathsf{add}(x_2, \mathsf{add}(x_1, b)) \mid x_2 \leftarrow p_2, x_1 \leftarrow p_1, b \leftarrow \nu \,|\!\}_{\mathrm{G}} \\
&\qquad \{ \text{ monad multiplication law } \} \\
&= \{\!|\, \mathsf{add}(x_2, b'') \mid x_2 \leftarrow p_2, b'' \leftarrow \{\!|\, \mathsf{add}(x_1, b) \mid x_1 \leftarrow p_1, b \leftarrow \nu \,|\!\}_{\mathrm{G}} \,|\!\}_{\mathrm{G}} \\
&\qquad \{ \text{ by definition } \} \\
&= \mathsf{distrAcc}_X(p_2, \mathsf{distrAcc}_X(p_1, \nu))
\end{aligned}$$

□

Having shown $\mathsf{distrAcc}_X$ to be measurable and symmetric we define the distributive law as its fold.

**Definition 4.3.3.** The distributive law between $\mathbf{B}$ and $\mathbf{G}$ is given by folding $\mathsf{distrAcc}_X$.

$$\begin{aligned}
&l_X : \mathbf{BG}X \to \mathbf{GB}X \\
&l_X(b) = \langle \mathsf{distrAcc}_X \rangle (b, \eta^{\mathbf{G}}_{\mathbf{B}X}(\varnothing))
\end{aligned}$$

Here $\eta^{\mathbf{G}}_{\mathbf{B}X}(\varnothing) \in \mathbf{GB}X$, the point process from which we can only sample the empty bag, is the initial argument into which the points sampled from the probability

measures in $b$ will be collected. $l_X$ is measurable by construction via fold.

In order to prove $l_X$ satisfies the required axioms it will be clearer to work with its monad comprehension form, which we can construct from the monad comprehension version of distrAcc.

$$
\begin{aligned}
&l_X : \mathbf{BG}X \to \mathbf{GB}X \\
&l_X(\llbracket p_1, \ldots, p_n \rrbracket) = \{\!\{ \mathsf{add}(x_1, \ldots \mathsf{add}(x_n, \varnothing) \ldots) \mid x_1 \leftarrow p_1, \ldots, x_n \leftarrow p_n \}\!\}_{\mathrm{G}} \\
&\qquad\qquad = \{\!\{ \llbracket x_1, \ldots, x_n \rrbracket \mid x_1 \leftarrow p_1, \ldots, x_n \leftarrow p_n \}\!\}_{\mathrm{G}}
\end{aligned}
$$

Rewriting our morphism in this form indeed confirms its operational behaviour we set out to obtain – a sample from $l_X(\llbracket \nu_1, \ldots, \nu_n \rrbracket)$ is the bag $\llbracket x_1, \ldots, x_n \rrbracket$ where each $x_i$ is a sample drawn from $\nu_i$. The sub-expression $x_1 \leftarrow p_1, \ldots, x_n \leftarrow p_n$ is the iterated application of the double strength dst, which is the product measure $\otimes_{i=1}^{n} p_i$. Furthermore the expression $\llbracket x_1, \ldots, x_n \rrbracket$ is equal to $q_n(x_1, \ldots, x_n)$ (and this is well-defined). From these two observations we can desugar the monad comprehension notation to arrive at the following equivalent definition of the distributive law.

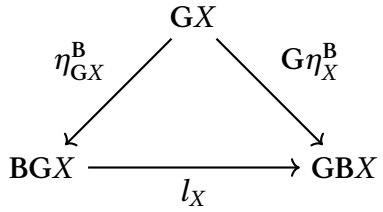
$$l_X(\llbracket p_1, \ldots, p_n \rrbracket) = \mathbf{G}(q_n)(\otimes_{i=1}^{n} p_i)$$

In the following we use the monad comprehension form of $l_X$ to prove that it satisfies all the required identities.

**Theorem 4.3.4.** *$l_X : \mathbf{BG}X \to \mathbf{GB}X$ is a distributive law between monads.*

*Proof.* We show that the two triangle and pentagon identities are satisfied by $l_X$. All the monad comprehensions in this proof are in the Giry monad $\mathbf{G}$ and so we omit the subscript for clarity.

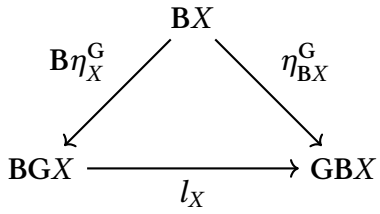
**Triangle I** Consider $p \in \mathbf{G}X$.




We have $\mathbf{G}(\eta^{\mathbf{B}}_X)(p) = \{\!\{ \eta^{\mathbf{B}}_X(x) \mid x \leftarrow p \}\!\} = \{\!\{ \llbracket x \rrbracket \mid x \leftarrow p \}\!\}$.

Along the other path, $l_X(\eta^{\mathbf{B}}_{\mathbf{G}X}(p)) = l_X(\llbracket p \rrbracket) = \{\!\{ \llbracket x \rrbracket \mid x \leftarrow p \}\!\}$.

**Triangle II** The diagram commutes for $\varnothing \in \mathbf{B}X$. Now take $\llbracket x_1, \ldots, x_n \rrbracket \in \mathbf{B}X$.



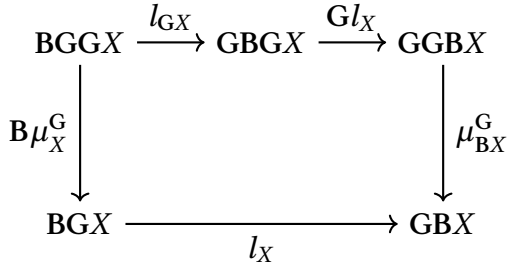

Going down the left edge we get the following.

$$\begin{aligned} l_X(\mathbf{B}(\eta^{\mathrm{G}}_X)(\llbracket x_1, \ldots, x_n \rrbracket)) &= l_X(\llbracket \eta^{\mathrm{G}}_X(x_1), \ldots, \eta^{\mathrm{G}}_X(x_n) \rrbracket) \\ &= \{\!\{ \llbracket y_1, \ldots, y_n \rrbracket \mid y_1 \leftarrow \eta^{\mathrm{G}}_X(x_1), \ldots, y_n \leftarrow \eta^{\mathrm{G}}_X(x_n) \}\!\} \end{aligned}$$

But by the monad laws for unit we have that $y_i = x_i$. Then the overall monad comprehension reduces to the one corresponding to the unit of $\mathbf{G}$.

$$\begin{aligned} l_X(\mathbf{B}(\eta^{\mathrm{G}}_X)(\llbracket x_1, \ldots, x_n \rrbracket)) &= \{\!\{ \llbracket x_1, \ldots, x_n \rrbracket \mid - \}\!\} \\ &= \eta^{\mathrm{G}}_{\mathbf{B}X}(\llbracket x_1, \ldots, x_n \rrbracket) \end{aligned}$$

**Pentagon I** Consider a bag of measures of measures $\llbracket \nu_1, \ldots, \nu_n \rrbracket \in \mathbf{BGG}X$ where $\nu_i \in \mathbf{GG}X$.




The top path of the pentagon yields the following transformation.

$$\begin{aligned} &\llbracket \nu_1, \ldots, \nu_n \rrbracket \in \mathbf{BGG}X \\ &\qquad \overset{l_{\mathbf{G}X}}{\longmapsto} \\ &\{\!\{ \llbracket p_1, \ldots, p_n \rrbracket \mid p_1 \leftarrow \nu_1, \ldots, p_n \leftarrow \nu_n \}\!\} \in \mathbf{GBG}X \end{aligned}$$

$$\xrightarrow{\mathbf{G}(l_X)}$$

$\{\!\{ l_X(\llbracket p_1, \ldots, p_n \rrbracket) \mid p_1 \leftarrow \nu_1, \ldots, p_n \leftarrow \nu_n \}\!\} \in \mathbf{GGB}X$

$= \{$ definition of $l_X$ $\}$

$\{\!\{ \{\!\{ \llbracket x_1, \ldots, x_n \rrbracket \mid x_1 \leftarrow p_1, \ldots, x_n \leftarrow p_n \}\!\} \mid p_1 \leftarrow \nu_1, \ldots, p_n \leftarrow \nu_n \}\!\} \in \mathbf{GGB}X$

$$\xrightarrow{\mu^{\mathrm{G}}_{\mathbf{B}X}}$$

$\{\!\{ \llbracket x_1, \ldots, x_n \rrbracket \mid p_1 \leftarrow \nu_1, \ldots, p_n \leftarrow \nu_n,\ \ x_1 \leftarrow p_1, \ldots, x_n \leftarrow p_n \}\!\} \in \mathbf{GB}X$

Along the bottom path we get the same result.

$\llbracket \nu_1, \ldots, \nu_n \rrbracket \in \mathbf{BGG}X$

$$\xrightarrow{\mathbf{B}(\mu^{\mathrm{G}}_X)}$$

$\llbracket \mu^{\mathrm{G}}_X(\nu_1), \ldots, \mu^{\mathrm{G}}_X(\nu_n) \rrbracket \in \mathbf{BG}X$

$$\xrightarrow{l_X}$$

$\{\!\{ \llbracket x_1, \ldots, x_n \rrbracket \mid x_1 \leftarrow \mu^{\mathrm{G}}_X(\nu_1), \ldots, x_n \leftarrow \mu^{\mathrm{G}}_X(\nu_n) \rrbracket \}\!\} \in \mathbf{GB}X$

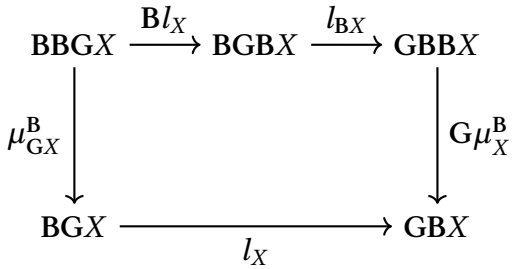
$= \{\ \mu^{\mathrm{G}}_X$ as a monad comprehension: $x_i \leftarrow \mu^{\mathrm{G}}_X(\nu_i) \ \rightsquigarrow\ p_i \leftarrow \nu_i, x_i \leftarrow p_i \}$

$\{\!\{ \llbracket x_1, \ldots, x_n \rrbracket \mid p_1 \leftarrow \nu_1, x_1 \leftarrow p_1,\ \ldots,\ p_n \leftarrow \nu_n,\ \ x_n \leftarrow p_n \}\!\} \in \mathbf{GB}X$

$= \{$ commutativity of $\mathbf{G}$ $\}$

$\{\!\{ \llbracket x_1, \ldots, x_n \rrbracket \mid p_1 \leftarrow \nu_1, \ldots, p_n \leftarrow \nu_n,\ \ x_1 \leftarrow p_1, \ldots, x_n \leftarrow p_n \}\!\} \in \mathbf{GB}X$

**Pentagon II** Consider a bag of bags of measures $\llbracket b_1, \ldots, b_n \rrbracket \in \mathbf{BBG}X$ where $b_i = \llbracket p_i^1, \ldots, p_i^{k_i} \rrbracket \in \mathbf{BG}X$ and $p_i^j \in \mathbf{G}X$. To make the presentation a little clearer we write $\llbracket \ldots, b_i, \ldots \rrbracket$ instead of $\llbracket b_1, \ldots, b_n \rrbracket$ below.



The top path of the pentagon yields the following transformation.

$$[\![\ldots,\ b_i,\ \ldots]\!] \in \mathbf{BBG}X$$

$$\xrightarrow{\mathbf{B}(l_X)}$$

$$[\![\ldots,\ l_X(b_i),\ \ldots]\!] \in \mathbf{BGB}X$$

$$= \{\ \text{definition of } l_X\ \}$$

$$\left[\!\!\left[\ldots,\ \{\!|[\![x_i^1,\ldots,x_i^{k_i}]\!] \mid x_1^{k_i} \leftarrow p_1^{k_i}|\!\},\ \ldots\right]\!\!\right] \in \mathbf{BGB}X$$

$$\xrightarrow{l_{\mathbf{B}X}}$$

$$\{\!|[\![\ldots,\ b_i',\ \ldots]\!] \mid \ldots,\ b_i' \leftarrow \{\!|[\![x_i^1,\ldots,x_i^{k_i}]\!] \mid x_i^{k_j} \leftarrow p_i^{k_j}|\!\},\ \ldots|\!\} \in \mathbf{GBB}X$$

$$= \{\ \text{by the monad laws}\ \}$$

$$\{\!|[\![\ldots,\ [\![x_i^1,\ldots,x_i^{k_i}]\!],\ \ldots]\!] \mid \ldots,\ x_i^{k_j} \leftarrow p_i^{k_j},\ \ldots|\!\} \in \mathbf{GBB}X$$

$$\xrightarrow{\mathbf{G}(\mu_X^{\mathbf{B}})}$$

$$\{\!|\mu_X^{\mathbf{B}}([\![\ldots,\ [\![x_i^1,\ldots,x_i^{k_i}]\!],\ \ldots]\!]) \mid \ldots,\ x_i^{k_j} \leftarrow p_i^{k_j},\ \ldots|\!\} \in \mathbf{GB}X$$

$$= \{\ \text{definition of } \mu_X^{\mathbf{B}}\ \}$$

$$\{\!|[\![\ldots,\ x_i^1,\ldots,x_i^{k_i},\ \ldots]\!] \mid \ldots,\ x_i^{k_j} \leftarrow p_i^{k_j},\ \ldots|\!\} \in \mathbf{GB}X$$

Along the bottom path we get the same result.

$$[\![\ldots,\ b_i,\ \ldots]\!] \in \mathbf{BBG}X$$

$$= \{\ \text{definition of } b_i\ \}$$

$$[\![\ldots,\ [\![p_i^1,\ldots,p_i^{k_i}]\!],\ \ldots]\!] \in \mathbf{BBG}X$$

$$\xrightarrow{\mu_{\mathbf{G}X}^{\mathbf{B}}}$$

$$[\![\ldots,\ p_i^1,\ldots,p_i^{k_i},\ \ldots]\!] \in \mathbf{BG}X$$

$$\xrightarrow{l_X}$$

$$\{\!|[\![\ldots,\ x_i^1,\ldots,x_i^{k_i},\ \ldots]\!] \mid \ldots,\ x_i^{k_j} \leftarrow p_i^{k_j},\ \ldots|\!\} \in \mathbf{GB}X$$

□

### 4.3.2 The distributive law as a measure

In the previous section we defined our distributive law $l_X : \mathbf{BG}X \to \mathbf{GB}X$. Consider a bag $b = [\![\nu_1, \ldots, \nu_n]\!] \in \mathbf{BG}X$. Then $l_X(b) \in \mathbf{GB}X$. By definition of the endofunctor $\mathbf{G}$, this means $l_X(b)$ is a measure on $\mathbf{B}X$. In other words, $l_X(b)$ is a function of type $\Sigma_{\mathbf{B}X} \to \mathbb{R}^+_\infty$. In this section we will discuss the details of this measure on our $\alpha$-sets.

Recall that we defined $l_X[\![\nu_1, \ldots, \nu_n]\!]$ to be the point process which samples a point each from $\nu_i \in \mathbf{G}X$ and returns the bag containing the $n$ points sampled. Then, when applied to the measurable set $\alpha_k^U$, it returns the probability of observing a bag such that $k$ of its elements belong to $U$. For example, when $n = 2$ and $k = 1$, $l[\![\nu_1, \nu_2]\!](\alpha_1^U)$ is the probability of either observing a point in $U$ coming from $\nu_1$ and not $\nu_2$ or $\nu_2$ and not $\nu_1$. This is given by $\nu_1(\overline{U})\nu_2(U) + \nu_1(U)\nu_2(\overline{U})$. In general, $l[\![\nu_1, \ldots, \nu_n]\!](\alpha_k^U)$ is the sum of $\binom{n}{k}$ terms where each term is the product of $k$ sets of $\nu_-(U)$ terms and $n-k$ sets of $\nu_-(\overline{U})$ terms. When $k > n$, $l[\![\nu_1, \ldots, \nu_n]\!](\alpha_k^U) = 0$ since it is impossible to sample any bags with more than $n$ elements. The sets $\alpha_k^U$ are a partition of the probability space.

$$\sum_{k=0}^{n} l_X[\![\nu_1, \ldots, \nu_n]\!](\alpha_k^U) = 1$$

The action of our distributive law can also alternatively be expressed using the pushforward along $\mathbf{G}$ of our measurable function $q_n : X^n \to \mathbf{B}_n X$ applied to the product measure of all the $\nu_i$'s, which is indeed the main intent behind the definition of $l_X$ as the fold of distrAcc.

$$l_X[\![\nu_1, \ldots, \nu_n]\!](\alpha_k^U) = \mathbf{G}(q_n)(\otimes_{i=1}^n \nu_i)$$

The following example calculation of $l[\![\nu_1, \nu_2]\!](\alpha_1^U)$ confirms this idea.

$$\begin{aligned} l[\![\nu_1, \nu_2]\!](\alpha_1^U) &= \mathbf{G}(q_2)(\nu_1 \otimes \nu_2)(\alpha_1^U) \\ &= (\nu_1 \otimes \nu_2)(q_2^{-1}[\alpha_1^U]) \\ &= (\nu_1 \otimes \nu_2)(U \times \overline{U} \cup \overline{U} \times U) \\ &= \nu_1(U)\nu_2(\overline{U}) + \nu_1(\overline{U})\nu_2(U). \end{aligned}$$

By definition, the sets $\alpha_k^U$ contain bags of varying cardinalities. In the following lemma we show that the measure $l[\![\nu_1, \ldots, \nu_n]\!]$ acts *only* on the subset of these sets

with cardinality $n$. This result is simple yet very useful in providing an intuitive understanding of the distributive law, helpful in simplifying calculations.

**Lemma 4.3.5.** *For $[\![ \nu_1, \ldots, \nu_n ]\!] \in \mathbf{BG}X$, $l[\![ \nu_1, \ldots, \nu_n ]\!](\alpha^X_m \cap W) = l[\![ \nu_1, \ldots, \nu_n ]\!](W)$ if $m = n$ and 0 if $m \neq n$. Here $X$ is the entire space, and so $\alpha^X_m$ is exactly the set of all bags in $X$ of cardinality $m$.*

*Proof.* Consider $l[\![ \nu_1, \ldots, \nu_n ]\!](\alpha^X_m)$. Using the definition of $l$ this probability can be expressed as the sum of products of a combination of $\nu_i(X)$ and $\nu_j(\overline{X})$ terms (where $i, j$ range from 1 to $n$). Unless $m = n$, each summand will contain at least one factor with $\nu_j(\overline{X}) = \nu_j(\varnothing) = 0$, nullifying the entire sum. It is only non-zero when $m = n$, representing the probability of observing $n$ points in the entire space, which has probability 1. The measure of any set $\alpha^X_n \cap W$ is then just the measure of $W$. □

### 4.3.3 The point process monad

Here we define a monadic structure on the endofunctor **GB** using the distributive law defined above.

**Definition 4.3.6.** The distributive law $l_X : \mathbf{BG}X \to \mathbf{GB}X$ induces a monad on **GB**. The unit $\eta^{\mathbf{GB}}$ is given by the composition of the respective units of **B** and **G**.

$$\eta^{\mathbf{GB}} = 1 \xrightarrow{\eta^{\mathbf{G}}\eta^{\mathbf{B}}} \mathbf{GB}$$

The multiplication is given by the multiplications of **B** and **G** along with the distributive law.

$$\mu^{\mathbf{GB}} = \mathbf{GBGB} \xrightarrow{\mathbf{G}l\mathbf{B}} \mathbf{GGBB} \xrightarrow{\mu^{\mathbf{G}}\mu^{\mathbf{B}}} \mathbf{GB}$$

Both **G** and **B** are strong monads (Theorem 3.6.6). We compose their strengths to define a strength for the point process monad. Define $t^{\mathbf{GB}}_{X,Y}$ to be the following composition.

$$t^{\mathbf{GB}}_{X,Y} = X \times \mathbf{GB}Y \xrightarrow{t^{\mathbf{G}}_{X,\mathbf{B}Y}} \mathbf{G}(X \times \mathbf{B}Y) \xrightarrow{\mathbf{G}(t^{\mathbf{B}}_{X,Y})} \mathbf{GB}(X \times Y)$$

**Lemma 4.3.7.** *The morphism $t^{\mathbf{GB}}_{X,Y} : X \times \mathbf{GB}Y \to \mathbf{GB}(X \times Y)$ is a strength for* **GB**.

*Proof.* Recall from Theorem 3.6.4 that since $t^{\mathrm{G}}$ is a strength it satisfies the equation $t^{\mathrm{G}}(x, p) = G(y \mapsto (x, y))(p)$. Furthermore we also have that $t^{\mathrm{B}}(x, b) = B(y \mapsto (x, y))(b)$. Consider $t^{\mathrm{GB}}(x, \nu)$ where $x \in X$ and $\nu \in \mathbf{GB}Y$.

$$\begin{aligned} t^{\mathrm{GB}}(x, \nu) &= G(t^{\mathrm{B}})(t^{\mathrm{G}}(x, \nu)) \\ &= G(t^{\mathrm{B}})(G(y \mapsto (x, y))(\nu)) \\ &= G(t^{\mathrm{B}})\{\!|(x, b) \mid b \leftarrow \nu|\!\}_{\mathrm{G}} \\ &= \{\!|t^{\mathrm{B}}(x, b) \mid b \leftarrow \nu|\!\}_{\mathrm{G}} \\ &= \{\!|\mathbf{B}(y \mapsto (x, y))(b) \mid b \leftarrow \nu|\!\}_{\mathrm{G}} \\ &= \mathbf{GB}(y \mapsto (x, y))(\nu) \end{aligned}$$

And so by the same result $t^{\mathrm{GB}}$ is the unique strength for **GB** due to it satisfying the equation $t^{\mathrm{GB}}(x, \nu) = \mathbf{GB}(y \mapsto (x, y))(\nu)$. □

From this follows our main theorem about the point process endofunctor **GB**.

**Theorem 4.3.8.** $(\mathbf{GB} : \mathbf{Meas} \to \mathbf{Meas}, \eta^{\mathrm{GB}}, \mu^{\mathrm{GB}}, t^{\mathrm{GB}})$ *is a strong monad on the category of measurable spaces.*

Note that although **G** and **B** are commutative monads, their composition **GB** is *not* commutative. We discuss this in more detail in the next section after calculating the bind of the point process monad, which makes it easier to find a quick counter-example.

### 4.3.4 The unit and bind of the point process monad

Having defined $\mathbf{GB} : \mathbf{Meas} \to \mathbf{Meas}$ as a monad, we now look at what the monadic structure gives us for composing and defining point processes.

The unit $\eta^{\mathrm{GB}}_X : X \to \mathbf{GB}X$ is defined as the horizontal composition

$$\eta^{\mathrm{GB}}_X = X \xrightarrow{\eta^{\mathrm{B}}_X} \mathbf{B}X \xrightarrow{\eta^{\mathrm{G}}_{\mathbf{B}X}} \mathbf{GB}X.$$

Given a point $x \in X$, $\eta^{\mathrm{GB}}_X(x)$ is the point process such that every sample from it is

the singleton bag consisting of $x$. Seen as a measure,

$$\eta_X^{\mathbf{GB}}(x) : \Sigma_{\mathbf{B}X} \to [0,1]$$
$$\eta_X^{\mathbf{GB}}(x) = W \in \Sigma_{\mathbf{B}X} \mapsto \begin{cases} 1, & \text{if } [\![x]\!] \in W \\ 0, & \text{otherwise.} \end{cases}$$

Next, the monad multiplication $\mu_X^{\mathbf{GB}} : \mathbf{GBGB}X \to \mathbf{GB}X$ is defined as the following composition.

$$\mu_X^{\mathbf{GB}} = \mathbf{GBGB}X \xrightarrow{\mathbf{G}(l_{\mathbf{B}X})} \mathbf{GGBB}X \xrightarrow{(\mu^{\mathbf{G}}\mu^{\mathbf{B}})_X} \mathbf{GB}X.$$

Although it is clear from the morphisms how the monad multiplication is supposed to work, we can get a better understanding of its operation by looking at the monadic bind of morphisms $f : X \to \mathbf{GB}Y$. That is, given $f : X \to \mathbf{GB}Y$ we need to construct $\gg\!=_{\mathbf{GB}}(f) : \mathbf{GB}X \to \mathbf{GB}Y$. Since this is a function from one probability measure to another, we can try to write it as the composition of the monadic bind (for the Giry monad) of certain maps (that is, given $g : X \to \mathbf{G}Y$, one has the bind $\gg\!=_{\mathbf{G}}(g) : \mathbf{G}X \to \mathbf{G}Y$).

In all, the monad multiplication can be decomposed into the morphisms $g_1$ and $g_2$

$$g_1 = \mathbf{B}X \xrightarrow{\mathbf{B}f} \mathbf{BGB}Y \xrightarrow{l_{\mathbf{B}Y}} \mathbf{GBB}Y \qquad\qquad g_2 = \mathbf{BB}Y \xrightarrow{\mu_X^{\mathbf{B}}} \mathbf{B}Y \xrightarrow{\eta_{\mathbf{B}Y}^{\mathbf{G}}} \mathbf{GB}Y$$

such that $\gg\!=_{\mathbf{GB}}(f)$ can be seen as the composition of their respective monadic binds.

$$\gg\!=_{\mathbf{GB}}(f) = \mathbf{GB}X \xrightarrow{\gg\!=_{\mathbf{G}}(g_1)} \mathbf{GBB}Y \xrightarrow{\gg\!=_{\mathbf{G}}(g_2)} \mathbf{GB}Y$$

Such a composition of maps is then easily converted to the corresponding monad comprehension notation in the $\mathbf{G}$ monad. Writing $\gg\!=_{\mathbf{GB}}$ as an infix operator, we get for $\alpha \in \mathbf{GB}X$ and $f : X \to \mathbf{GB}Y$,

$$\alpha \gg\!=_{\mathbf{GB}} f \in \mathbf{GB}Y$$
$$\alpha \gg\!=_{\mathbf{GB}} f = \{\!|\, \mu_Y^{\mathbf{B}}(y) \mid b \leftarrow \alpha,\ y \leftarrow (l_{\mathbf{B}X} \circ \mathbf{B}(f))(b) \,|\!\}_{\mathbf{G}}.$$

Having written the bind as a $\mathbf{G}$-comprehension, we can analyse its operational

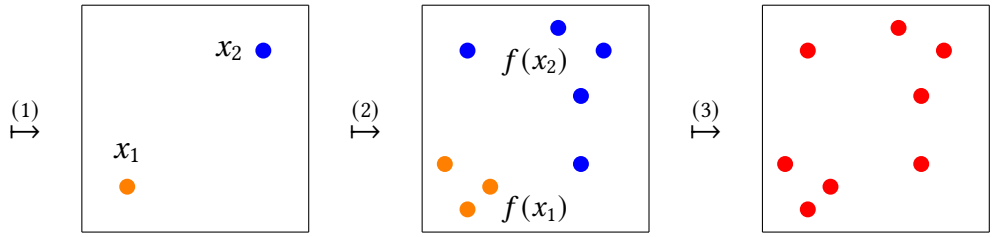



Figure 4.1: Sampling from a composite process.

semantics via the operational semantics of the Giry monad. Recall that in a Giry monad comprehension, $x \leftarrow p$ is to be read as "sample an $x$ from the probability measure $p$". And so we can see the process of sampling a bag of points from $\alpha \gg\!=_{\mathrm{GB}} f \in \mathbf{GB}Y$ as the following process. First sample a bag of points $b \in \mathbf{B}X$ from $\alpha$. Say $b = [\![x_1, \ldots, x_n]\!]$. Next, to each element $x_i$ apply $f : X \to \mathbf{GB}Y$, giving us a bag of point processes in $Y$ ($\mathbf{BGB}Y$). Now apply the distributive law $l_{\mathbf{B}X}$ to this bag in order to sample a bag of bags $y \in \mathbf{BB}Y$. But recall from the definition of the distributive law (Section 4.3.1) that a sample from $l_X(b')$ for some $b' = [\![p_1, \ldots, p_n]\!] \in \mathbf{BG}Z$ is simply a bag of points where each point is a sample from one of the $p_i \in \mathbf{G}Z$. In our case $p_i = f(x_i)$, and so each bag in $y$ is a sampled bag of points from $f(x_i)$ for each $x_i$. Finally we take the union of all the bags in $y$ and return this bag of $Y$ as the sampled bag.

We can therefore summarise the process of sampling a bag of points in $Y$ from $\alpha \gg\!=_{\mathrm{GB}} f \in \mathbf{GB}Y$ as the following three-step procedure, illustrated in Figure 4.1.

1. Sample a bag of points from $\alpha$.
2. Each point $x_i$ in this bag produces a point process $f(x_i)$ from which we sample a bag of points in $Y$.
3. A sample of points from the overall point process is the union of these bags of points in $Y$.

This intuition is quite powerful as it allows us to declaratively program with point processes by being able to define them simply by how they must be simulated. We make extensive use of this intuition in the next section.

**Non-commutativity.** It is well-known that the Giry and bag monads are commutative, but this does not extend to the point process monad. Consider the following two point processes $\gamma_1, \gamma_2 \in \mathbf{GB1}$ on the unit type. Define $\gamma_1$ to be the point process

which returns either the empty bag or the singleton bag $[\![\star]\!]$ with equal probability 1/2, and let $\gamma_2$ be the point process which returns the bag $[\![\star, \star]\!]$ with probability 1. Then consider the following two point processes formed by composing $\gamma_1$ and $\gamma_2$ with the bind $\gg=_{\mathbf{GB}}$.

$$\Delta_1, \Delta_2 \in \mathbf{GB1}$$
$$\Delta_1 = \{\!|\star \mid \star \leftarrow \gamma_1, \star \leftarrow \gamma_2|\!\}_{\mathbf{GB}}$$
$$\Delta_2 = \{\!|\star \mid \star \leftarrow \gamma_2, \star \leftarrow \gamma_1|\!\}_{\mathbf{GB}}$$

Then drawing a sample from $\Delta_1$ consists of first drawing a bag from $\gamma_1$. When this is the empty bag the sampling process is completed and the overall sample drawn from $\Delta_1$ will be the empty bag. When it is the singleton $[\![\star]\!]$, we draw a sample once from $\gamma_2$, which is going to be the bag $[\![\star, \star]\!]$. In all we sample either the empty bag or the two-element bag with probability 1/2 each. In the case of $\Delta_2$ we start by drawing the two-element bag from $\gamma_2$ with probability 1. Then we draw two independent samples from $\gamma_1$ and return their union. This results in the final sample being either the empty bag with probability 1/4, the singleton bag with probability 1/2 (either the first sample from $\gamma_1$ was the singleton and the second empty or vice-versa), or the two-element bag with probability 1/4. Clearly then $\Delta_1$ is not equal to $\Delta_2$.

**Theorem 4.3.9.** *The point process monad is not commutative.*

## 4.4 Monadic point process constructions

Now that we have given a monadic structure to the point process monad **GB** and shown how to define primitive recursive functions on finite bags via the fold construction, we can combine these approaches to give examples of compositional constructions of point processes. To this end we show how to view probability distributions on countable sets, how to define standard operations on point processes in a way that their measurable structure is preserved by construction, and finally how to define more complex point processes from constituent smaller point processes.

We implement all these constructions in LazyPPL [Dash et al., 2023], a Haskell library for writing probabilistic programs, which can be found in Appendix A. The idea is that these constructions become much easier to write when viewed

compositionally and functionally.

### 4.4.1 Probability distributions as point processes

**Probability distributions on the naturals.** As a first example, we describe probability distributions on the natural numbers as point processes on the singleton space $\mathbf{1} = \{\star\}$, based on the observation that a bag of singletons is a natural number. In other words, we have the isomorphism $\mathbf{GB1} \cong \mathbf{G}\mathbb{N}$ due to $\mathbf{B1} \cong \mathbb{N}$.

Any probability distribution $d \in \mathbf{G}\mathbb{N}$ (so that $\sum_{i=0}^{\infty} d_i = 1$) can be presented as a point process $\underline{d} \in \mathbf{GB1}$ where we observe $k$ copies of $\star$ with probability $d_k$.

$$\underline{d}\left(\alpha_k^{\{\star\}}\right) := d_k$$

Given $d \in \mathbf{G}\mathbb{N}$, we have that $\underline{d} = \mathbf{G}(\mathsf{natToBag})(d)$ where $\mathsf{natToBag} : \mathbb{N} \to \mathbf{B1}$ is the isomorphism.

**Building compound probability distributions on $\mathbb{N}$.** Using our monad we define compound distributions on $\mathbb{N}$ as point processes on the unit type.

A compound probability distribution is the probability distribution of the sum of a *random* number of independent identically-distributed random variables, i.e., where the number of terms to be added is itself a random variable. For example, given a random variable $N \sim \text{Poisson}(\Lambda)$ and i.i.d. random variables $X_i$, the random variable $Y = \sum_{i=0}^{N} X_i$ forms a compound Poisson distribution.

Recall the behaviour of the monadic bind $\gg\!=_{\mathbf{GB}}$ on countable sets described in Section 4.3.4. Let $N \in \mathbf{G}\mathbb{N}$ be a Poisson distribution with rate $\Lambda$, for example. Then $\underline{N} \in \mathbf{GB1}$ is the corresponding point process as outlined above. Similarly let $X \in \mathbf{G}\mathbb{N}$ represent our i.i.d. random variables $X_i$, which we also write as a point process $\underline{X} \in \mathbf{GB1}$. Then we can express the compound distribution above as the following point process written as a monad comprehension.

$$\begin{aligned}&\gamma \in \mathbf{GB1}\\ &\gamma = \{\!|\star \mid \star \leftarrow \underline{N}, \star \leftarrow \underline{X}|\!\}_{\mathbf{GB}}\end{aligned}$$

Following the operational behaviour of $\gg\!=_{\mathbf{GB}}$ we can see that we get exactly what we set out to define. We first sample some Poisson number of points from $\underline{N}$ and

then for *each point* we sample a bag of points from $\underline{X}$. These points are all combined into a bag and returned.

Note that due to the non-commutativity of the point process monad, it is important that we first sample a bag of points from $\underline{N}$, and only then for each point in the bag we sample a bag of points from $\underline{X}$. Writing the monad comprehension in the other order would define the random variable $Y' = \sum_{i=0}^{X} N_i$, which is not the same.

### 4.4.2 Point process operations

In point process theory it is often useful to approximate models by considering well-known point processes to which certain operations have been applied. Here we consider three common point process operations which we define as morphisms on $\mathbf{GB}X$.

**Superposition.** Let $\gamma_1, \gamma_2 \in \mathbf{GB}X$ be two arbitrary point processes. The superposition of these point processes is a point process $\gamma \in \mathbf{GB}X$ such that a sample drawn from it is the (multiplicity-preserving) union of a sample of bags drawn from $\gamma_1$ and $\gamma_2$ each.

This operation is defined as the functorial action of $\uplus : \mathbf{B}X \times \mathbf{B}X \to \mathbf{B}X$ applied to the samples from $\gamma_1$ and $\gamma_2$, which in turn can be obtained from the double strength $\mathrm{dst}_{X,Y} : \mathbf{G}X \times \mathbf{G}Y \to \mathbf{G}(X \times Y)$. This can be written concisely as a $\mathbf{G}$ monad comprehension.

$$\boxplus : \mathbf{GB}X \times \mathbf{GB}X \to \mathbf{GB}X$$
$$\boxplus(\gamma_1, \gamma_2) = \{\!\{ b_1 \uplus b_2 \mid b_1 \leftarrow \gamma_1, b_2 \leftarrow \gamma_2 \}\!\}_{\mathbf{G}}$$

**Thinning a point process.** Thinning is an operation applied to the points of an underlying point process, where the points are independently thinned (removed) according to some probabilistic rule. Given some point process $\gamma \in \mathbf{GB}X$ and some thinning rule $t : X \to \mathbf{GB}X$ such that $t(x)$ probabilistically returns either $[\![x]\!]$ or $\varnothing$, we can use the $\mathbf{GB}$ monad to build the thinned point process $\gamma' \in \mathbf{GB}X$ as below.

$$\gamma' \in \mathbf{GB}X$$
$$\gamma' = \{\!\{ x' \mid x \leftarrow \gamma, x' \leftarrow t(x) \}\!\}_{\mathbf{GB}}$$

Seen operationally, we first sample a bag of points from $\gamma$. Then for each point $x$ in that bag we sample a bag from $t(x)$. This is either the singleton bag containing $x$, in which case it appears in the final sample, or it is the empty bag, in which case it is omitted. The result is the union of all these bags, which are either singletons or empty, giving us the thinned sample. It is possible that a bag sampled from $\gamma$ will have the same point occurring more than once. Even in this case each point will be thinned independently of each other, since for each $x_i$ we sample a new bag $x'_i$.

**Displacing a point process.** Displacement is an operation applied to the points of an underlying point process, where the points are independently randomly displaced (translated) according to some distribution.

Suppose we are given some distribution $\Delta \in \mathbf{G}\mathbb{R}$ according to which we displace our points. We lift $\Delta$ to construct the point process $\Delta' = \mathbf{G}(\eta^{\mathbf{B}}_{\mathbb{R}})(\Delta) \in \mathbf{GB}\mathbb{R}$, the samples of which are always singleton bags containing the random displacement amount. For $\gamma \in \mathbf{GB}\mathbb{R}$ we simulate the displaced point process $\gamma' \in \mathbf{GB}\mathbb{R}$ by sampling an $x$ from $\gamma$, a displacement distance $d$ from $\Delta'$, and then returning the displaced point.

$$\gamma' \in \mathbf{GB}\mathbb{R}$$
$$\gamma' = \{\!|\, x + d \mid x \leftarrow \gamma, d \leftarrow \mathbf{G}(\eta^{\mathbf{B}}_{\mathbb{R}})(\Delta) \,|\!\}_{\mathbf{GB}}$$

The argument for why the points are displaced independently of each other is the same as that for thinning.

### 4.4.3 Constructing point processes

Following probability distributions and standard operations, we now define various point processes using **GB** comprehensions.

**Poisson point processes.** A homogeneous Poisson point process with rate $\Lambda$ on the unit square $[0,1]^2$ can be simulated by first sampling a random number of points from the Poisson distribution, and then uniformly distributing each of these points across $[0,1]^2$ [Last and Penrose, 2017]. Let $\underline{N}_\Lambda \in \mathbf{GB1}$ be the Poisson distribution with rate $\Lambda$ seen as a point process, which we will use to determine the random number of points in our sample. Now we need to define a point process $U_2 \in \mathbf{GB}[0,1]^2$, samples from which are singleton bags consisting of a point

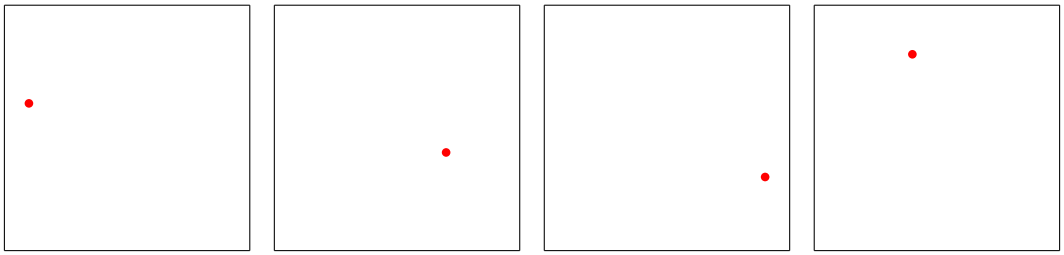


Figure 4.2: Samples from $U_2 \in \mathbf{GB}[0,1]^2$.

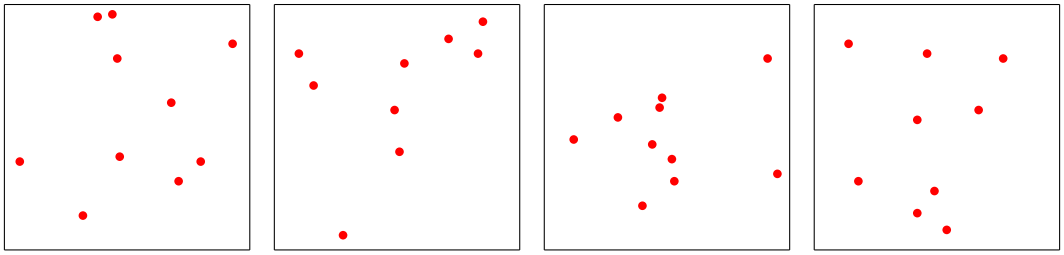


Figure 4.3: Samples from Poisson point process $\pi_{10} \in \mathbf{GB}[0,1]^2$ with rate $\Lambda = 10$.

uniformly distributed in $[0,1]^2$. This can be defined by independently sampling two points from the uniform distribution $\mathsf{unif} \in \mathbf{G}[0,1]$. Figure 4.2 contains samples from $U_2$.

$$
\begin{aligned}
&U_2 \in \mathbf{GB}[0,1]^2 \\
&U_2 = \{\!|\, \eta^{\mathbf{B}}_{[0,1]^2}((x,y)) \mid x \leftarrow \mathsf{unif}, y \leftarrow \mathsf{unif} \,|\!\}_{\mathbf{G}}
\end{aligned}
$$

The Poisson point process $\pi_\Lambda$ can then be written as the following comprehension.

$$
\begin{aligned}
&\pi_\Lambda \in \mathbf{GB}[0,1]^2 \\
&\pi_\Lambda = \{\!|\, (x,y) \mid \star \leftarrow \underline{N}_\Lambda, (x,y) \leftarrow U_2 \,|\!\}_{\mathbf{GB}}
\end{aligned}
$$

As before, we first sample a bag of points from $\underline{N}_\Lambda$ and then for each point we sample a random point in the unit square, and finally we combine all of these points into a bag. Figure 4.3 contains samples from this point process.

**Clustered point processes.** Clustered point processes are useful in modelling phenomena which involve multiple points spawning from individual seeds, such as clusters of stars in galaxies. Informally, a clustered point process is anything built using the monadic bind $\gamma_1 \gg\!=_{\mathbf{GB}} (x \mapsto \gamma_2(x))$, where $\gamma_1$ is a point process for the initial seeds, and $\gamma_2(x)$ is a point process that grows from each seed, where the

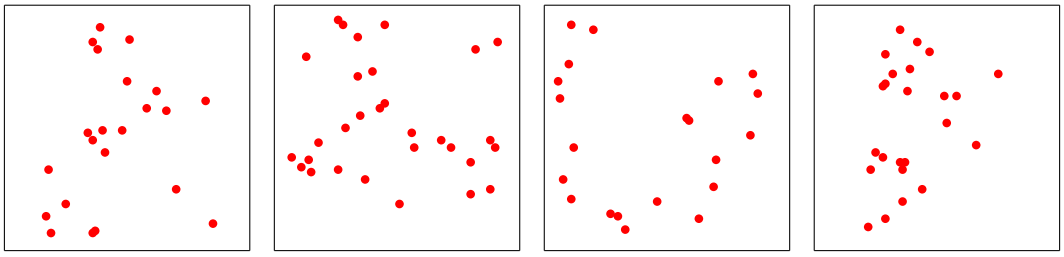


Figure 4.4: Samples from the Thomas cluster process with $\Lambda_1 = 6$, $\Lambda_2 = 3$, $\sigma^2 = 0.1$.

location $x$ of a given seed may be a parameter.

A well-known example of such a point process on the Euclidean plane is the Thomas Cluster Process [Thomas, 1949]. Draws from a Thomas process are bags with clusters of points, which are simulated by first sampling seeds from a Poisson point process with rate $\Lambda_1$. For each seed we sample a Poisson number of offsprings with rate $\Lambda_2$. Each of these new points is distributed normally with variance $\sigma^2$ around the center of the cluster. We can easily define such a point process using monad comprehensions. We already have the Poisson point process $\pi_{\Lambda_1} \in \mathbf{GB}[0,1]^2$ and the Poisson distribution $\underline{N}_{\Lambda_2} \in \mathbf{GB1}$ with rate $\Lambda_2$ as previously defined. Using $\mathsf{normal} : \mathbb{R}\times\mathbb{R}^+ \to \mathbf{G}\mathbb{R}$ we can define $\mathsf{norm} = \mathbf{G}(\eta^B_{\mathbb{R}}) \circ \mathsf{normal} : \mathbb{R}\times\mathbb{R}^+ \to \mathbf{GB}\mathbb{R}$ to pick the coordinates of the offsprings and obtain the cluster process $\tau \in \mathbf{GB}[0,1]^2$.

$$\tau \in \mathbf{GB}[0,1]^2$$
$$\tau = \{\!|\, (x,y) \mid (x',y') \leftarrow \pi_{\Lambda_1}, \star \leftarrow \underline{N}_{\Lambda_2}, x \leftarrow \mathsf{norm}(x',\sigma^2), y \leftarrow \mathsf{norm}(y',\sigma^2) \,|\!\}$$

Figure 4.4 contains simulated samples from $\tau$. This example is simple, but already illustrates that we can use the monad to quickly and clearly compose point processes to build complex statistical models.

**Matern hard-core point processes.** In the clustering example just now we saw that the monadic bind of the **GB** monad is useful in producing point processes exhibiting attraction between points. Often we may also want our points to exhibit some repulsion. An example of this is the simulation of locations of trees – although the presence of trees in one area makes it more likely to have more of them in the surrounding region, it is also the case that these trees won't exist arbitrarily close to each other (since they compete for resources and require space to grow).

Point processes which model these kinds of scenarios where two points need

to have a minimum fixed distance between each other are known as hard-core point processes (the idea behind the naming being that each point has a hard impenetrable core around it). Many different kinds of hard-core point processes have been studied, each with their own respective simulation algorithms which guarantee these minimum fixed distances. One such point process that we will look at is called the Matern hard-core point process, introduced in [Matérn, 1960]. More specifically, we will look at the Type I Matern hard-core point process. A draw from this point process proceeds in two steps. First we sample a bag of points from a homogeneous Poisson point process with some rate. Next, for each point in the bag, we keep it if there is no other point within some distance $r$ of it, and discard it otherwise. This thinning step is, in a sense, not too trivial to define from scratch in a way that immediately guarantees its measurability. We can, however, define such a measurable map by making use of monad comprehensions in the bag monad.

And so let us define such a function $\mathsf{thin}_r : \mathbf{B}[0,1]^2 \to \mathbf{B}[0,1]^2$ which removes all points in a bag that are within distance $r$ of each other. For this we first define the function $\mathsf{filter}_p : \mathbf{B}X \to \mathbf{B}X$ which is parameterised by a measurable boolean predicate $p : X \to \mathsf{Bool}$ and removes the elements of a bag not satisfying $p$. Given $p : X \to \mathsf{Bool}$ we can lift it to the measurable function $\hat{p} : X \to \mathbf{B1}$ which is the composition of $p$ with the function which sends True to $[\![\star]\!]$ and False to $\varnothing$.

This lifting enables us to define $\mathsf{filter}_p$ as a monad comprehension where elements not satisfying $p$ do not get included due to $\hat{p}$ evaluating to the empty bag.

$$\begin{aligned}&\mathsf{filter}_p : \mathbf{B}X \to \mathbf{B}X\\&\mathsf{filter}_p(b) = \{\!|\, x \mid x \leftarrow b,\ \star \leftarrow \hat{p}(x) \,|\!\}_{\mathbf{B}}\end{aligned}$$

Define the function $t_r : X \times \mathbf{B}X \to \mathbf{B}X$ to send $(x, b)$ to $[\![x]\!]$ if there is no $x' \in b$ such that $|x - x'| < r$ and $\varnothing$ otherwise. This is again measurably definable as a fold by, for example, constructing a new bag of booleans $b'$ where the boolean indicates if $x$ is not within distance $r$ of $x'$, and then returning True if there is not a single False in $b'$, and False otherwise. It is a standard functional programming exercise using folds and its implementation can be found in Appendix A.

Now given a bag $b \in \mathbf{B}[0,1]^2$, for each $x \in b$ we can obtain a new bag $b_{\overline{x}}$ which doesn't contain $x$ – this can be written as a filter with the predicate checking equality with $x$. $[0,1]^2$ is a standard Borel space and so the equality predicate

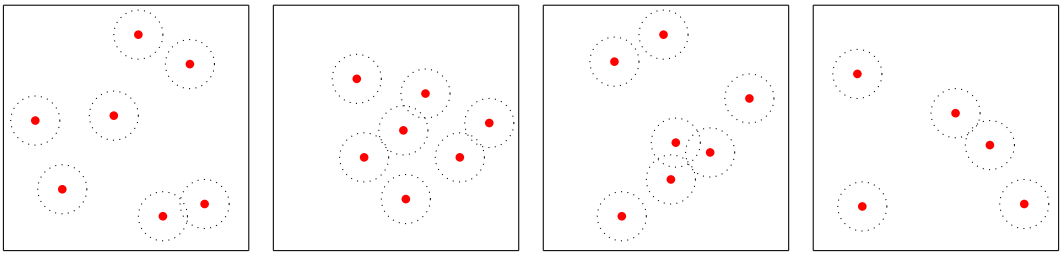


Figure 4.5: Samples from the Matern hard-core point process with $\Lambda = 15$, $r = 0.1$.

$\mathsf{eq} : [0,1]^2 \times [0,1]^2 \to \mathsf{Bool}$ is measurable. Then we can apply $t_r$ to each such pair, and finally take the union of all the resultant singleton bags. This is what we want $\mathsf{thin}_r : \mathbf{B}[0,1]^2 \to \mathbf{B}[0,1]^2$ to do, and the complete definition can be written as the following monad comprehension.

$$\begin{aligned}&\mathsf{thin}_r : \mathbf{B}[0,1]^2 \to \mathbf{B}[0,1]^2\\&\mathsf{thin}_r(b) = \{\!|x \mid y \leftarrow b, x \leftarrow t_r(y, \mathsf{filter}_{\neq y}(b))|\!\}_{\mathbf{B}}\end{aligned}$$

Finally the Matern point process can be defined as $\mathbf{G}(\mathsf{thin}_r)$ applied to the Poisson point process $\pi_\Lambda \in \mathbf{GB}[0,1]^2$. Figure 4.5 contains samples from this point process where we sample from a Poisson point process with rate $\Lambda = 15$ and filter out points within a distance of $r = 0.1$ of each other (the dotted circles have radius $r = 0.1$). Note that the expected number of points in this point process is much lower than 15 because we end up having to throw away many points due to the higher likelihood of them being close to each other in the unit square and therefore overlapping within the restricted radius.

$$\begin{aligned}&\mathsf{matern} \in \mathbf{GB}[0,1]^2\\&\mathsf{matern} = \mathbf{G}(\mathsf{thin}_r)(\pi_\Lambda)\end{aligned}$$

This construction of the Matern hard-core point process is measurably valid at every step, all thanks to the fold construction and point process monad ensuring measurability of our functions, as well as respecting the bag structure. We can now, for example, use a Matern point process to generate seeds of a stochastic model, where each seed can give rise to nearby offsprings. Writing out such a point process by hand can very quickly get tedious and ensuring everything is measurable even more so. And so, with these examples, it is clear that the combination of the **GB** monad along with primitive recursion on bags by folding gives us a rich and

expressive language for defining useful and meaningful transformations on point processes which are measurable by construction.

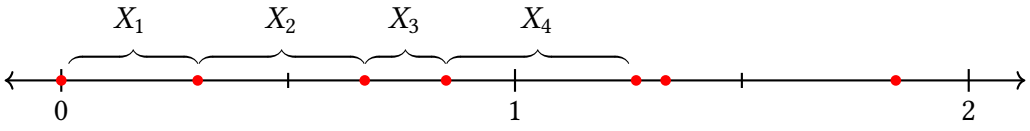
## 4.5 Towards infinite point processes

So far we have only considered *finite* point processes. In the study of point processes it is not uncommon to present point processes as random infinite bags, a prime example being the Poisson point process. In this work we considered a homogeneous Poisson point process with rate $\lambda$ as a random finite bag on (say) the unit interval $[0, 1]$ where a draw from this process, restricted to a measurable subset $A \subseteq [0, 1]$, contains a Poisson random number of points (more specifically the random number is $N \sim \mathsf{Pois}(\lambda \times |A|)$ where $|A|$ is the Lebesgue measure of $A$).

There is an equivalent characterisation of this point process where it is defined on the positive real line, usually given as a generative process [Last and Penrose, 2017]. In this view the real line is traditionally seen as a time axis and the points are times at which some random event occurs. This way, rather than talking about the number of points in some subset of the real line, we instead focus on the "inter-arrival times" of the events which are the (random) time differences between successive events. In particular, in the case of the Poisson point process, the inter-arrival times are independent real random variables distributed according to the exponential distribution with parameter $\lambda$. A random variable $X$ has this distribution, denoted $X \sim \mathsf{Exp}(\lambda)$, when it has the following probability density function.

$$f(x; \lambda) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0, \\ 0 & \text{if } x < 0 \end{cases}$$

For example, below is a draw from a Poisson point process on the real line by sampling inter-arrival times $X_i$ from $\mathsf{Exp}(10)$. It is the equivalent of the standard definition of a Poisson point process defined on a fixed space (which here is the closed interval $[0, 2]$) with rate $\Lambda = 5$.



And so this generative process can be translated into our setting as a random infinite bag of reals (in the case of the Poisson point process there won't be any overlapping points). We showed in Section 3.8 that it is unclear if the measurable space of infinite bags on reals $(\mathbf{B}_\infty\mathbb{R}, \Sigma_{\mathbf{B}_\infty\mathbb{R}})$ is standard Borel. Even though it is entirely possible to work with measurable spaces that are not standard Borel, the fact that they do not guarantee any nice algebraic properties (such as closure under countable products and coproducts) makes it inconvenient to use them as sub-programs of larger programs.

There is, however, a good alternative. Instead of working with random bags, we can work with random infinite sequences, also known as random streams. A stream of $X$ is simply the countable product $X^{\mathbb{N}}$, and this construction also exists in the category of measurable spaces. Streams are very well behaved objects and have been studied extensively in mathematics and computer science. Furthermore with streams we have the advantage that $(X^{\mathbb{N}}, \Sigma_{X^{\mathbb{N}}})$ is standard Borel when $(X, \Sigma_X)$ is standard Borel (due to closure under countable products).

In the case of the Poisson point process as a generative process, we can consider draws from the process to be random streams where the components of the sampled stream are the locations of the points on the positive real line. This is easily generalised to the 2-dimensional (and $n$-dimensional) case too, where we can work in polar coordinates and therefore form a lexicographic ordering of the points sampled. This allows us to view these points as elements of an infinite stream.

In the remainder of this chapter we will look at random streams. There is a stream endofunctor $\mathrm{S} : \mathbf{Meas} \to \mathbf{Meas}$ in the category of measurable spaces which also happens to be monadic. As a result, we will look at how the Giry monad composes with this monad. We will see that there actually doesn't exist a distributive law $\kappa_X : \mathrm{SG}X \to \mathrm{GS}X$, but there is a *weak* distributive law. A weak distributive law is another such morphism $\mathrm{SG}X \to \mathrm{GS}X$ but it needs to satisfy one fewer axiom than a full distributive law. More specifically, the axiom that need not be satisfied is the unit axiom pertaining to the unit of $\mathrm{G}$.

$$\mathrm{G} \xrightarrow{\eta^{\mathrm{G}}\mathrm{S}} \mathrm{GS} = \mathrm{G} \xrightarrow{\mathrm{S}\eta^{\mathrm{G}}} \mathrm{SG} \xrightarrow{\kappa} \mathrm{GS}.$$

### 4.5.1 Measurable streams and the stream monad

**Definition 4.5.1.** Given a set $X$, define $\mathrm{S}(X)$ to be the set of all streams on $X$. That is, $\mathrm{S}(X)$ is the infinite product space $X^{\mathbb{N}}$.

$$\mathrm{S}(X) = X \times X \times X \times \cdots = X^{\mathbb{N}}$$

The functorial action of $f : X \to Y$ is given by point-wise application.

$$\begin{aligned} &\mathrm{S}(f) : \mathrm{S}X \to \mathrm{S}Y \\ &\mathrm{S}(f)(x_1, x_2, \dots) = (f(x_1), f(x_2), \dots) \end{aligned}$$

In contrast with finite bags, lifting the space of streams on $X$ to the measurable space of streams is more straightforward due to **Meas** having countable products.

**Definition 4.5.2** (The measurable space of streams)**.** The endofunctor $\mathrm{S} : \mathbf{Meas} \to \mathbf{Meas}$ sends the measurable space $(X, \Sigma_X)$ to $(\mathrm{S}X, \Sigma_{\mathrm{S}X})$, the measurable space of streams on $X$ with the $\sigma$-algebra $\Sigma_{\mathrm{S}X}$ given by

$$\Sigma_{\mathrm{S}X} = \sigma\left(\{U_1 \times \cdots \times U_n \times \mathrm{S}X \mid n \in \mathbb{N}, U_i \in \Sigma_X\}\right).$$

That is, the $\sigma$-algebra is generated by the set of all infinite-dimensional rectangles formed from the product of elements in $\Sigma_X$ where all but a finite number of sets are $X$. This is the standard definition of the $\sigma$-algebra of an infinite product, and is analogous to the definition of countable products on topological spaces. Note that unlike in the topological setting, the infinite product $\prod_{i\in\mathbb{N}} U_i$ *is* a measurable subset of $\mathrm{S}X$ (given that $U_i \in \Sigma_X$ for all $i \in \mathbb{N}$). This is because $\sigma$-algebras are closed under countable unions, and so this set is simply the countable intersection of our generating sets $\bigcap_{i\in\mathbb{N}} X^i \times U_i \times \mathrm{S}X$.

The functorial action of $f : (X, \Sigma_X) \to (Y, \Sigma_Y)$ remains the same as in **Set**.

$$\begin{aligned} &\mathrm{S}(f) : (\mathrm{S}X, \Sigma_{\mathrm{S}X}) \to (\mathrm{S}Y, \Sigma_{\mathrm{S}Y}) \\ &\mathrm{S}(f)(x_1, x_2, \dots) = (f(x_1), f(x_2), \dots) \end{aligned}$$

That $\mathrm{S}(f)$ is measurable is clear from the fact that it maps inverse images of sets in

$\Sigma_{SY}$ to $\Sigma_{SX}$.

$$\mathrm{S}(f)^{-1}\left[\prod_{i=1}^{n} U_i \times Y^{\mathbb{N}}\right] = \prod_{i=1}^{n} f^{-1}[U_i] \times (f^{-1}[Y])^{\mathbb{N}}$$

With the stream endofunctor $\mathrm{S} : \mathbf{Meas} \to \mathbf{Meas}$ we can now talk about infinite point processes on $X$ as random streams $\gamma \in \mathbf{G}\mathrm{S}X$. That is, $\gamma : \Sigma_{\mathrm{S}X} \to [0, 1]$ is a measure on streams.

In the category of sets the stream endofunctor admits a strong monad structure, where the unit returns the constant stream and the multiplication returns the diagonal of a stream of streams. In functional programming this monad is often referred to as the reader monad.

$$\begin{aligned}
&\eta^{\mathrm{S}}_X : X \to \mathrm{S}X \\
&\eta^{\mathrm{S}}_X(x) = (x, x, x, \dots)
\end{aligned}$$

$$\begin{aligned}
&\mu^{\mathrm{S}}_X : \mathrm{S}(\mathrm{S}X) \to \mathrm{S}X \\
&\mu^{\mathrm{S}}_X(s_1, s_2, \dots) = (\pi_1(s_1), \pi_2(s_2), \dots)
\end{aligned}$$

$$\begin{aligned}
&t_{X,Y} : X \times \mathrm{S}Y \to \mathrm{S}(X \times Y) \\
&t_{X,Y}(x, s) = \big((x, \pi_1(s)), (x, \pi_2(s)), \dots\big)
\end{aligned}$$

It is not difficult to show that the monad laws are satisfied. For the left (resp. right) unit law we take any stream $(x_1, x_2, \dots) \in \mathrm{S}X$ and apply $\eta^{\mathrm{S}}_{\mathrm{S}X}$ (resp. $\mathrm{S}\eta^{\mathrm{S}}_X$) to get a stream of streams. This can be seen as a 2-dimensional infinite array, where the values are constant along the rows (resp. columns). Then applying $\mu^{\mathrm{S}}_X$ gives us the diagonal, which is equal to the stream we started with. For the associativity law we consider an element $s \in \mathrm{SS}\mathrm{S}X$ as an infinite 3-dimensional array. The two ways of applying $\mu^{\mathrm{S}}$ to this stream correspond to taking diagonal plane sections of the array. The order of application doesn't matter as we always get the diagonal at the end.

These maps are the same in the measurable setting, for which the only additional item needed is proofs that maps are measurable.

**Theorem 4.5.3.** $(\mathrm{S} : \mathbf{Meas} \to \mathbf{Meas}, \eta^{\mathrm{S}}, \mu^{\mathrm{S}}, t^{\mathrm{S}})$ *is a strong monad in the category of measurable spaces.*

*Proof.* Each of the three maps is measurable due to the universal mapping property of the countable product $\mathrm{S}(X \times Y) = (X \times Y)^{\mathbb{N}}$.

- Unit: $\eta^{\mathrm{S}}_X = \times_{i \in \mathbb{N}} \mathrm{id}_X : X \to \mathrm{S}X$.
- Multiplication: $\mu^{\mathrm{S}}_X = \times_{i \in \mathbb{N}} \pi_i \circ \pi_i : \mathrm{S}(\mathrm{S}X) \to \mathrm{S}X$.
- Strength: $t^{\mathrm{S}}_{X,Y} = \times_{i \in \mathbb{N}} \mathrm{id}_X \times \pi_i : X \times \mathrm{S}Y \to \mathrm{S}(X \times Y)$.

□

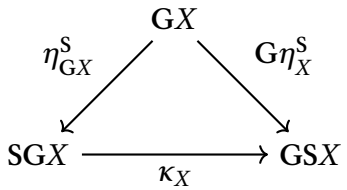
### 4.5.2 Non-existence of a distributive law

Having shown $\mathrm{S} : \mathbf{Meas} \to \mathbf{Meas}$ to be a monad, we now look at the composition of $\mathrm{G}$ with $\mathrm{S}$ and answer the question of whether $\mathrm{G} \circ \mathrm{S}$ also admits a monad via a distributive law.

In this section we give a negative answer to this question by showing that there is no natural transformation $\kappa_X : \mathrm{SG}X \to \mathrm{GS}X$ simultaneously satisfying two of the axioms required by a distributive law. In the next section we show that there is still a $\kappa_X : \mathrm{SG}X \to \mathrm{GS}X$ satisfying three of the axioms, making it a weak distributive law between $\mathrm{S}$ and $\mathrm{G}$.

**Theorem 4.5.4.** *There is no distributive law* $\kappa_X : \mathrm{SG}X \to \mathrm{GS}X$.

*Proof.* We show that no such distributive law can exist by demonstrating that there is no natural transformation from $\mathrm{SG}X$ to $\mathrm{GS}X$ simultaneously satisfying the axioms pertaining to the unit and multiplication of the stream monad. We first consider the unit axiom, from which we will derive some requirements on $\kappa_X : \mathrm{SG}X \to \mathrm{GS}X$.



Take some $p \in \mathrm{G}X$ at the vertex of the triangle. Going down the left-hand side we get the constant stream $(p, p, \dots)$; then finally going along the base we get $\kappa_X(p, p, \dots)$. We can rewrite the transformation along the right edge using monad comprehension notation, giving us $\{\!|(z, z, \dots) \mid z \leftarrow p|\!\}$. That is, our distributive law $\kappa$ at the very least needs to satisfy $\kappa_X(p, p, \dots) = \{\!|(z, z, \dots) \mid z \leftarrow p|\!\}$.

Now we use the naturality of $\kappa$ to make some additional deductions about its action on certain specific streams of probability measures $s \in \mathrm{SG}X$. In the following, let $X$ be the 4-element set $\{a, b, c, d\}$ and let $Y$ be the 2-element set $\{x, y\}$, each equipped with the powerset $\sigma$-algebra. We will define two specific functions $f_1, f_2 : X \to Y$ and consider the naturality squares they form with $\kappa$. Define $f_1, f_2 : X \to Y$ as follows.

$$f_1 = \begin{matrix} a \mapsto x \\ b \mapsto x \\ c \mapsto y \\ d \mapsto y \end{matrix} \qquad f_2 = \begin{matrix} a \mapsto x \\ b \mapsto y \\ c \mapsto x \\ d \mapsto y \end{matrix}$$

Note that since $X$ and $Y$ are finite sets, the elements of $\mathrm{G}X$ and $\mathrm{G}Y$ are probability *distributions*, and so they can be described entirely by writing out the ratios of the respective probabilities of individual elements. And so we shall write, as an example, $(1 : 1 : 1 : 1) \in \mathrm{G}X$ as the probability distribution where $a, b, c$, and $d$ are distributed with equal probability, and $(3 : 1) \in \mathrm{G}Y$ as the distribution where the probability of observing $x$ is $3/4$ and $y$ is $1/4$. Furthermore, since we have probability distributions, the functorial action of $\mathrm{G}$ also becomes simple to express (for example, $\mathrm{G}(f_1)(1 : 2 : 3 : 4) = ((1 + 2) : (3 + 4)) = (3 : 7)$, and $\mathrm{G}(f_2)(1 : 2 : 3 : 4) = ((1 + 3) : (2 + 4)) = (4 : 6)$).

With this notation, we define two specific distinct probability distributions on $X$, call them $\phi_1$ and $\phi_2 \in \mathrm{G}X$. We pick the ratios of the probabilities of $\phi_i = (r_{a,i} : r_{b,i} : r_{c,i} : r_{d,i}) \in \mathrm{G}X$ such that

$$r_{a,i} + r_{b,i} = r_{c,i} + r_{d,i}$$
$$r_{a,i} + r_{c,i} = r_{b,i} + r_{d,i}.$$

An example solution to this pair of equations gives us $\phi_1 = (1 : 2 : 2 : 1)$ and $\phi_2 = (2 : 1 : 1 : 2)$. We defined these distributions specifically like this so that $\mathrm{G}(f_1)(\phi_1) = \mathrm{G}(f_2)(\phi_1) = \mathrm{G}(f_1)(\phi_2) = \mathrm{G}(f_2)(\phi_2) = (1 : 1) \in \mathrm{G}Y$. Define $\psi = (1 : 1) \in \mathrm{G}Y$ – this is the uniform distribution on $Y$.

Now we pick a stream of distributions $s \in \mathrm{SG}X$ such that $s_i$ is either equal to $\phi_1$ or $\phi_2$ with the additional condition that $s$ is not equal to the constant streams $(\phi_1, \phi_1, \dots)$ and $(\phi_2, \phi_2, \dots)$. Consider the two paths that this stream $s$ takes in the naturality squares for $f_1$ and $f_2$.

$$\begin{array}{ccc} \mathrm{SG}X & \xrightarrow{\mathrm{SG}f_i} & \mathrm{SG}Y \\ {\scriptstyle \kappa_X}\downarrow & & \downarrow{\scriptstyle \kappa_Y} \\ \mathrm{GS}X & \xrightarrow[\mathrm{GS}f_i]{} & \mathrm{GS}Y \end{array}$$

We calculate $\mathrm{SG}f_1(s)$ and $\mathrm{SG}f_2(s)$ and note that they both evaluate to the constant stream of $\psi$, the uniform distribution of $x$ and $y$ (due to our choice of $f_1$, $f_2$, and $s$).

$$\mathrm{SG}f_1(s) = \mathrm{SG}f_2(s) = \big(\psi, \psi, \dots\big) \in \mathrm{SG}Y$$

We apply $\kappa_Y$ to this constant stream and apply the unit axiom to obtain the following random stream.

$$\kappa_Y(\mathrm{SG}f_i(s)) = \kappa_Y(\psi, \psi, \dots) = \{\!|(z, z, \dots) \mid z \leftarrow \psi|\!\}_{\mathrm{G}} \in \mathrm{GS}Y$$

This is the infinite stream which is either constantly $x$ or constantly $y$ with probablility 1/2 each. Now along the bottom paths of the respective naturality squares we get that both $\mathrm{GS}f_1(\kappa_X(s))$ and $\mathrm{GS}f_2(\kappa_X(s))$ need to be equal to $\{\!|(z, z, \dots) \mid z \leftarrow \psi|\!\}_{\mathrm{G}}$.

First consider the equality $\mathrm{GS}f_1(\kappa_X(s)) = \{\!|(z, z, \dots) \mid z \leftarrow \psi|\!\}_{\mathrm{G}} \in \mathrm{GS}Y$. From this we get that a sample from $\kappa_X(s)$ is a stream such that applying $\mathrm{S}f_1$ to it gives either the constant $x$ or constant $y$ stream, and these outcomes are equally likely. Since $f_1$ maps $a, b$ to $x$ and $c, d$ to $y$, we can conclude that a sample from $\kappa_X(s)$ is either a stream consisting of only the values $a$ and $b$, or only $c$ and $d$ (and it is impossible to have any other mix of values).

Now we can perform similar reasoning for $f_2$, where we have $\mathrm{GS}f_2(\kappa_X(s)) = \{\!|(z, z, \dots) \mid z \leftarrow \psi|\!\}_{\mathrm{G}} \in \mathrm{GS}Y$. $f_2$ maps $a, c$ to $x$ and $b, d$ to $y$, and so we also obtain the fact that a stream sampled from $\kappa_X(s)$ will either contain only $a$ and $c$ or only $b$ and $d$ values. Combining the information from these two constraints we obtain that a sample from $\kappa_X(s)$ is either the constant stream $(a, a, \dots)$ or

$(d, d, \dots)$ with equal probability. As a Giry monad comprehension,

$$\kappa_X(s) = \{\!|(e, e, \dots) \mid e \leftarrow (1 : 0 : 0 : 1)|\!\}_G \in \mathrm{GS}X.$$

Now consider the distributive law axiom below, relating to $\mu^{\mathrm{S}}$.

$$\begin{array}{ccccc}
\mathrm{SSG}X & \xrightarrow{\mathrm{S}\kappa_X} & \mathrm{SGS}X & \xrightarrow{\kappa_{\mathrm{G}X}} & \mathrm{GSS}X \\
\Big\downarrow{\scriptstyle \mu^{\mathrm{S}}_{\mathrm{G}X}} & & & & \Big\downarrow{\scriptstyle \mathrm{G}\mu^{\mathrm{S}}_X} \\
\mathrm{SG}X & & \xrightarrow{\qquad\kappa_X\qquad} & & \mathrm{GS}X
\end{array}$$

Take $t = (t_1, t_2, \dots) \in \mathrm{SSG}X$ where $t_i \in \mathrm{SG}X$ is a stream containing $\phi_1 \in \mathrm{G}X$ at position $i$ and $\phi_2 \in \mathrm{G}X$ everywhere else (in other words $t$ contains $\phi_1$ along the diagonal and $\phi_2$ everywhere else).

First we go down the bottom path. Applying $\mu^{\mathrm{S}}_{\mathrm{G}X}$ gives us all the elements along the diagonal, which is $(\phi_1, \phi_1, \dots) \in \mathrm{SG}X$. Applying $\kappa_X$ to this gives us $\{\!|(e, e, \dots) \mid e \leftarrow \phi_1|\!\}_G \in \mathrm{GS}X$.

Along the top path we first apply $\mathrm{S}\kappa_X$ to obtain

$$\begin{aligned}
\mathrm{S}\kappa_X(t_1, t_2, \dots) &= (\kappa_X(t_1), \kappa_X(t_2), \dots) \in \mathrm{SGS}X \\
&= (\tau, \tau, \dots) \qquad (\text{where } \tau = \{\!|(e, e, \dots) \mid e \leftarrow (1 : 0 : 0 : 1)|\!\}_G).
\end{aligned}$$

Apply $\kappa_{\mathrm{G}X}$ to this to get $\{\!|(f, f, \dots) \mid f \leftarrow \tau|\!\}_{\mathrm{G}} \in \mathrm{GSS}X$. Finally we apply $\mathrm{G}\mu^{\mathrm{S}}_X$ to get $\{\!|\mu^{\mathrm{G}}_X(f, f, \dots) \mid f \leftarrow \tau|\!\}_{\mathrm{G}} \in \mathrm{GS}X$. Simplify this using the monad laws.

$$\begin{aligned}
&\{\!|\mu^{\mathrm{G}}_X(f, f, \dots) \mid f \leftarrow \tau|\!\}_{\mathrm{G}} \\
&\quad = \{\!|\mu^{\mathrm{G}}_X(f, f, \dots) \mid f \leftarrow \{\!|(e, e, \dots) \mid e \leftarrow (1 : 0 : 0 : 1)|\!\}_G|\!\}_{\mathrm{G}} \\
&\quad = \{\!|\mu^{\mathrm{G}}_X((e, e, \dots), (e, e, \dots), \dots) \mid e \leftarrow (1 : 0 : 0 : 1)|\!\}_{\mathrm{G}} \\
&\quad = \{\!|(e, e, \dots) \mid e \leftarrow (1 : 0 : 0 : 1)|\!\}_{\mathrm{G}}
\end{aligned}$$

This is not equal to the expression we got along the bottom path,

$$\{\!|(e, e, \dots) \mid e \leftarrow \phi_1|\!\}_G = \{\!|(e, e, \dots) \mid e \leftarrow (1 : 2 : 2 : 1)|\!\}_G.$$

And so $\kappa_X : \mathbf{SG}X \to \mathbf{GS}X$ cannot be a distributive law. □

### 4.5.3 A weak distributive law

Even though there is no distributive law between $\mathbf{S}$ and $\mathbf{G}$, there is a very natural candidate morphism $\mathbf{SG}X \to \mathbf{GS}X$ which happens to satisfy the remaining axioms of a distributive law, making it a weak distributive law. We now outline its construction.

Consider $\kappa_X : \mathbf{SG}X \to \mathbf{GS}X$. This is a map from streams of probability measures $(\nu_1, \nu_2, \dots) \in \mathbf{SG}X$ to a probability measure on streams $\kappa_X(\nu_1, \nu_2, \dots) \in \mathbf{GS}X$. When analysed as a sampling procedure, we can consider a stream sampled from $\kappa_X(\nu_1, \nu_2, \dots)$ as a stream $(x_1, x_2, \dots)$ where each $x_i$ is a sample from $\nu_i$. This is similar to what we did in the finite bags case, and such a transformation obeys the remaining axioms of a distributive law in a clean syntactic manner. First we introduce relevant machinery to define such a measurable morphism taking streams of measures to a measure on streams.

One needs to be careful when defining a map into $\mathbf{GS}X$. In general, the Giry monad does *not* preserve all limits, with a simple example being finite products. There is a result in probability theory known as the Kolmogorov extension theorem which nicely allows defining maps into $\mathbf{GS}X$ by defining probability measures on finite dimensional distributions $\mathbf{G}X^n$ (such that some compatibility conditions hold), but it is not a purely measure theoretic theorem and requires some topological assumptions on the measures we consider (namely inner regular measures). This is fine when working with Polish spaces, and there is a useful categorical description of the result, sometimes called the Bochner extension theorem, which says that the Giry monad on Polish spaces preserves projective limits [Bochner, 1955, Choksi, 1958, Rao, 1971], allowing us to easily give maps into infinite product measures.

Although these extension theorems make it possible to write definitions elegantly and concisely, their lack of generality to all measurable spaces renders them unsuitable for our needs. Instead, we make use of a similar result, but one which is purely measure theoretic, known as the Ionescu-Tulcea extension theorem [Ionescu-Tulcea, 1949]. In the literature this theorem is often stated using *Markov kernels*, which are simply measurable maps $X \to \mathbf{G}Y$, and are frequently written as the equivalent map $X \times \Sigma_Y \to [0, 1]$. This map needs to be measurable in $X$ for fixed $W \in \Sigma_Y$, and for fixed $x \in X$ needs to be a probability measure on $Y$.

We first introduce new notation for the composition of a particular family of Markov kernels, which helps simplify the statement of the theorem to follow. Let $m_1 : X \to \mathbf{G}Y$ and $m_2 : X \times Y \to \mathbf{G}Z$ be two Markov kernels. Then we define the composite kernel $m_1 \otimes m_2 : X \to \mathbf{G}Z$ as the following map.

$$
\begin{aligned}
&m_1 \otimes m_2 : X \to \mathbf{G}Z \\
&m_1 \otimes m_2 = x \mapsto \{\!|\, z \mid y \leftarrow m_1(x), z \leftarrow m_2(x, y) \,|\!\}_{\mathbf{G}}
\end{aligned}
$$

**Theorem 4.5.5** (Ionescu-Tulcea's extension)**.** *Let $\{X_i, \Sigma_{X_i}\}_{i=1}^{\infty}$ be a sequence of measurable spaces and let $\nu \in \mathbf{G}X_1$ be a probability measure on $X_1$. Denote by $\vec{X}_j$ the $j$-product $\prod_{i=1}^{j} X_j$. Now let $m_j : \vec{X}_j \to \mathbf{G}X_{j+1}$ be a family of Markov kernels. Then there exists a unique probability measure $\iota \in \mathbf{G}(\prod_{i=1}^{\infty} X_i)$ such that $\iota(W \times \mathbf{S}X) = \overline{m}_j(W)$ where $W \in \Sigma_{\vec{X}_j}$ and $\overline{m}_j \in \mathbf{G}(\vec{X}_j)$ is the probability measure on $\vec{X}_j$ defined by the iterated composition of $\nu$ with the Markov kernels $\{m_i : \vec{X}_i \to \mathbf{G}X_{i+1}\}_{i=1,\ldots,j-1}$.*

$$
\begin{aligned}
&\overline{m}_j : \Sigma_{\vec{X}_j} \to [0, 1] \\
&\overline{m}_j = \begin{cases} \nu & \text{if } j = 1 \\ \left\{\!\!\left|\, z \;\middle|\; x \leftarrow \nu, z \leftarrow \left(\bigotimes_{i=1}^{j-1} m_i\right)(x) \,\right|\!\!\right\}_{\mathbf{G}} & \text{otherwise} \end{cases}
\end{aligned}
$$

Apart from the topological assumptions, the key difference between Kolmogorov's and Ionescu-Tulcea's theorems is in the kind of random streams that they allow us to construct. Using Ionescu-Tulcea we can easily build random streams where the values of later components can only depend on previous components, and for this reason it is widely used in the theory of Markov chains. Kolmogorov, on the other hand, allows us to define more complex random streams (and even uncountable product measures) where the dependencies between the random components can be much more complicated (making it more suitable for defining things like Brownian motion, for example).

Using Ionescu-Tulcea we can now define a simple infinite product measure where none of the components of the product depend on any previous components.

**Lemma 4.5.6.** *Let $(X, \Sigma_X)$ and $\{(Y_i, \Sigma_{Y_i})\}_{i=1}^{\infty}$ be measurable spaces and suppose there is a family of maps $\{f_i : X \to \mathbf{G}Y_i\}_{i=1}^{\infty}$. Then there is a family of maps $\{g_i : \mathbf{G}(\prod_{i=j}^{\infty} Y_j) \to \mathbf{G}Y_i\}_{i=1}^{\infty}$ and a mediating morphism $p : X \to \mathbf{G}(\prod_{i=j}^{\infty} Y_j)$ such*

*that* $f_i = p \circ g_i : X \to \mathbf{G}Y_i$.

*Proof.* We first construct the mediating morphism using Ionescu-Tulcea's extension theorem. In other words we we are defining a map from $x \in X$ to some $\nu_x \in \mathbf{G}(\prod_{i=1}^{\infty} Y_i)$, which we will construct. To make use of the theorem we need a family $m_j : Y_1 \times \cdots \times Y_j \to \mathbf{G}Y_{j+1}$ of Markov kernels and a probability measure $m_0 \in \mathbf{G}Y_1$. $m_0$ can be taken to be $f_1(x) \in \mathbf{G}Y_1$. Next each $m_j : Y_1 \times \cdots \times Y_j \to \mathbf{G}Y_{j+1}$ can be the Markov kernel which ignores its input and simply returns $f_{j+1}(x)$. From this we get the unique product measure $p_x \in \mathbf{G}(\prod_{i=1}^{\infty} Y_i)$. Since all the $m_j$ Markov kernels ignore their input, their results are just independent random variables. From this we can calculate $p_x(W_1 \times \cdots \times W_n \times \prod_{i=n+1}^{\infty} Y_i)$ to be $f_1(x)(W_1) \times \cdots \times f_n(x)(W_n)$.

Let $Y = \prod_{i=1}^{\infty} Y_i$. Now we need to show that the map $x \mapsto p_x : X \to \mathbf{G}Y$ is measurable. We will write it as the equivalent $(x, W) \mapsto p_x(W) : X \times \Sigma_Y \to [0, 1]$ and prove that this is a Markov kernel. For this we first show that for all $W \in \Sigma_Y$, $x \mapsto p_x(W)$ is measurable in $X$. Define the set $D = \{W \in \Sigma_Y \mid x \mapsto p_x(W) \text{ is measurable}\} \subseteq \Sigma_Y$. Showing $D = \Sigma_Y$ would imply that indeed for all $W \in \Sigma_Y$ the map $x \mapsto p_x(W)$ is a measurable map. And so we are left with showing $\Sigma_Y \subseteq D$.

For this we will make use of the $\pi$-$\lambda$ theorem (Theorem 3.2.14). Define another set $D' = \{W_1 \times \cdots \times W_n \times \prod_{j=n+1}^{\infty} Y_j \mid W_i \in \Sigma_{Y_i}\}$. Note that $\sigma(D') = \Sigma_Y$ (by definition of the product $\sigma$-algebra). The $\pi$-$\lambda$ theorem states that if $D'$ is a $\pi$-system and $D$ is a $\lambda$-system containing $D'$, then $(\Sigma_Y =)\ \sigma(D') \subseteq D$.

To see that $D'$ is a $\pi$-system (that is, it is closed under pairwise intersections) take any two elements $U_1 \times \cdots \times U_m \times \prod_{j=m+1}^{\infty} Y_j$ and $V_1 \times \cdots \times V_n \times \prod_{j=n+1}^{\infty} Y_j$. The intersection of these two sets is just component-wise intersection. Each component will still be a measurable set in the respective $\Sigma_{Y_i}$ (since they are closed under intersections), and there are only a finite number of components that are not equal to the entire set $Y_i$. And so the intersection is also a generating set in $D'$.

To see that $D$ is a $\lambda$-system we need to show that it contains the empty set (trivial), is closed under complements, and is closed under countable unions of pairwise disjoint sets. Suppose $W \in D$, meaning $x \mapsto p_x(W)$ is measurable. Then we can also conclude that $x \mapsto 1 - p_x(W)$ is measurable, meaning $x \mapsto p_x(\overline{W})$ is measurable. And so $\overline{W} \in D$. The argument for closure under countable union of pairwise disjoint sets is the same, where the measure applied to the union becomes a

sum of measures applied to each set, $p_x(\bigcup_j W_j) = \sum_j p_x(W_j)$ and its measurability immediately follows from the assumption that the $W_j$ sets are elements of $D$.

Finally we need to show that $D' \subseteq D$. Take $U_1 \times \cdots \times U_n \times \prod_{j=n+1}^{\infty} Y_i \in D'$. Consider $x \mapsto p_x(U_1 \times \cdots \times U_n \times \prod_{j=n+1}^{\infty} Y_i)$. From Ionescu-Tulcea this is just $x \mapsto f_1(x)(U_1) \times \cdots \times f_n(x)(U_n)$. Each $f_i : X \to \mathbf{G}Y_i$ is given to be measurable, and so the total map is also measurable. And so $U_1 \times \cdots \times U_n \times \prod_{j=n+1}^{\infty} Y_i \in D$, giving us $D' \subseteq D$. From this we conclude $D = \Sigma_Y$ and so $x \mapsto p_x(W)$ is measurable for all $W \in \Sigma_Y$.

Finally we note that $W \mapsto p_x(W) : \Sigma_Y \to [0,1]$ is a measure for all $x \in X$ (this follows from the Ionescu-Tulcea construction). And so $(x, W) \mapsto p_x(W) : X \times \Sigma_Y \to [0,1]$ is a Markov kernel, and so $x \mapsto p_x$ is measurable, completing our proof. □

In order to define a map $\kappa_X : \mathbf{SG}X \to \mathbf{GS}X$ it now suffices to use the lemma above and give maps $\kappa_X^i : \mathbf{SG}X \to \mathbf{G}X$. We then recover $\kappa_X : \mathbf{SG}X \to \mathbf{GS}X$ as the mediating morphism. This leads us to our definition of our candidate weak distributive law.

**Definition 4.5.7** (Weak distributive law)**.** Let $\kappa_X^i = \pi_i : \mathbf{SG}X \to \mathbf{G}X$ be a family of maps. Then define $\kappa_X : \mathbf{SG}X \to \mathbf{GS}X$ to be the mediating morphism obtained from Lemma 4.5.6. From this we get that $\kappa_X(s)(W_1 \times \cdots \times W_n \times \mathbf{S}X) = \pi_1(s)(W_1) \times \cdots \times \pi_n(s)(W_n)$.

We can informally write this definition using monad comprehension notation for a clearer picture.

$$\kappa_X(s) = \{\!|(x_1, x_2, \dots) \mid x_1 \leftarrow \pi_1(s), x_2 \leftarrow \pi_2(s), \dots|\!\}_{\mathbf{G}}$$

We emphasise that this is only informal, and not a precise definition, which is due to the presence of the infinite "…", for which we haven't given a formal semantics. This intuition is nonetheless useful to have.

Before proceeding to the main result, we record a useful well-known lemma from measure theory regarding the equality of measures defined on $\sigma$-algebras generated by some family of sets. It will be useful in the upcoming theorem for showing the equality of the measures obtained along the paths of the commutative diagrams we consider.

**Lemma 4.5.8.** *Let $(X, \Sigma_X)$ be a measurable space where $\Sigma_X = \sigma(\mathcal{F})$ with $\mathcal{F}$ being some family of subsets of $X$, closed under finite intersections. That is, $\mathcal{F}$ is a $\pi$-system. Then two measures $\nu_1, \nu_2 : \Sigma_X \to \mathbb{R}^+_\infty$ are equal if and only if they agree on the generating sets $W \in \mathcal{F}$. In other words, $\nu_1 = \nu_2 \iff \nu_1(W) = \nu_2(W)\ \forall W \in \mathcal{F}$.*

This lemma follows from the $\pi$-$\lambda$ theorem. In our case, the $\sigma$-algebra for $\mathrm{S}X$ is generated by sets of the form $W_1 \times \cdots \times W_n \times \mathrm{S}X$ where $W_i \in \Sigma_X$ and $n \in \mathbb{N}$, and so two measures $\nu_1, \nu_2 \in \mathrm{GS}X$ are equal when they agree on all sets $W_1 \times \cdots \times W_n \times \mathrm{S}X$.

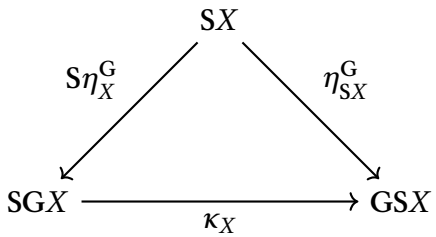
**Theorem 4.5.9.** *$\kappa_X : \mathrm{SG}X \to \mathrm{GS}X$ is a weak distributive law between* S *and* G.

*Proof.* We show that the triangle and pentagon identities are satisfied by $\kappa_X$. For each diagram we first formally prove it commutes by using the definition of $\kappa_X$ in Definition 4.5.7. This will be followed by an informal argument, using the monad comprehension equivalent of $\kappa_X$, which is morally the same as the formal definition. We include it in order to make it clear that this proof is almost entirely syntactic, and can be shown using just the properties of monads alone.

We remark that in the informal proofs we are reasoning about entire random streams in one go, whereas formally this goes through by considering the application of our resultant measures in $\mathrm{GS}X$ to all the finite prefix measurable sets $W = W_1 \times \cdots \times W_n \times \mathrm{S}X \in \Sigma_{\mathrm{S}X}$ (where $W_i \in \Sigma_X$) of the random streams (due to Lemma 4.5.8).

All the monad comprehensions in this proof are in the Giry monad G and so we omit the subscript for clarity. We sometimes use $(x_i)_{i=1}^{\infty}$ as shorthand for the infinite sequence $(x_1, x_2, \dots)$.

**Triangle** Take $(x_i, x_2, \dots) \in \mathrm{S}X$.



Going down the right edge and applying $W = W_1 \times \cdots \times W_n \times \mathrm{S}X \in \Sigma_{\mathrm{S}X}$ we get

$$\eta^{\mathrm{G}}_{\mathrm{S}X}(x_1, x_2, \dots)(W) = \delta_{(x_1,x_2,\dots)}(W) = \prod_{i=1}^{n} \delta_{x_i}(W_i).$$

And along the left edge

$$\kappa_X(\mathrm{S}\eta^{\mathrm{G}}_X(x_1, x_2, \dots))(W) = \kappa_X(\delta_{x_1}, \delta_{x_2}, \dots)(W) = \prod_{i=1}^{n} \delta_{x_i}(W_i).$$

We can look at the computation informally via our monad comprehension too. Given $(x_1, x_2, \dots) \in \mathrm{S}X$, we have the following.

$$\begin{aligned}\kappa_X(\mathrm{S}(\eta^{\mathrm{G}}_X)(x_1, x_2 \dots)) &= \kappa_X(\eta^{\mathrm{G}}_X(x_1), \eta^{\mathrm{G}}_X(x_2), \dots) \\ &= \{\!\{(y_1, y_2, \dots) \mid y_1 \leftarrow \eta^{\mathrm{G}}_X(x_1), y_2 \leftarrow \eta^{\mathrm{G}}_X(x_2), \dots\}\!\}\end{aligned}$$

But by the monad laws for unit we get that $y_i = x_i$. Then the overall monad comprehension reduces to the one corresponding to the unit of $\mathbf{G}$.

$$\begin{aligned}&= \{\!\{(x_1, x_2, \dots) \mid -\}\!\} \\ &= \eta^{\mathrm{G}}_{\mathrm{S}X}(x_1, x_2, \dots)\end{aligned}$$

**Pentagon I** Consider a stream of measures of measures $(\nu_i)_{i=1}^{\infty} \in \mathbf{SGG}X$ where $\nu_i \in \mathbf{GG}X$.

$$\begin{array}{ccccc} \mathbf{SGG}X & \xrightarrow{\kappa_{\mathbf{G}X}} & \mathbf{GSG}X & \xrightarrow{\mathbf{G}\kappa_X} & \mathbf{GGS}X \\ \Big\downarrow{\scriptstyle \mathrm{S}\mu^{\mathrm{G}}_X} & & & & \Big\downarrow{\scriptstyle \mu^{\mathrm{G}}_{\mathrm{S}X}} \\ \mathbf{SG}X & & \xrightarrow[\kappa_X]{} & & \mathbf{GS}X \end{array}$$

Along the left edge we get

$$\begin{aligned}\kappa_X(\mathrm{S}\mu^{\mathrm{G}}_X(\nu_i)_{i=1}^{\infty})(W) &= \kappa_X(\mu^{\mathrm{G}}_X(\nu_i))_{i=1}^{\infty}(W) \\ &= \prod_{i=1}^{\infty} \mu^{\mathrm{G}}_X(\nu_i)(W_i)\end{aligned}$$

$$= \prod_{i=1}^{\infty} \int_{p_i \in \mathrm{G}X} p_i(W_i)\, \nu_i(\mathrm{d}p_i).$$

Going the other way evaluate an integral to arrive at the same answer.

$$\begin{aligned}
&\mu^{\mathrm{G}}_{\mathrm{S}X}(\mathbf{G}\kappa_X(\kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty}))(W) \\
&\quad = \int_{\mathrm{GS}X} \mathrm{ev}_W \, \mathrm{d}(\mathbf{G}\kappa_X(\kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty})) \\
&\quad = \int_{\mathrm{GS}X} \mathrm{ev}_W \, \mathrm{d}((\kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty}) \circ \kappa_X^{-1}) \\
&\qquad\qquad \{ \text{ by change of variables } \} \\
&\quad = \int_{\mathrm{SG}X} \mathrm{ev}_W \circ \kappa_X \, \mathrm{d}(\kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty}) \\
&\qquad\qquad \{ \text{ introducing the explicit variable } (p_i)_{i=1}^{\infty} \in \mathrm{SG}X \} \\
&\quad = \int_{(p_i)_{i=1}^{\infty} \in \mathrm{SG}X} (\mathrm{ev}_W \circ \kappa_X)(p_i)_{i=1}^{\infty} (\kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty} \, \mathrm{d}(p_i)_{i=1}^{\infty}) \\
&\qquad\qquad \{ \text{ definition of } \mathrm{ev}_W \} \\
&\quad = \int_{(p_i)_{i=1}^{\infty} \in \mathrm{SG}X} \kappa_X(p_i)_{i=1}^{\infty}(W) \ (\kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty} \, \mathrm{d}(p_i)_{i=1}^{\infty}) \\
&\qquad\qquad \{ \text{ definition of } \kappa_X \} \\
&\quad = \int_{(p_i)_{i=1}^{\infty} \in \mathrm{SG}X} \prod_{i=1}^{n} p_i(W_i) \ (\kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty} \, \mathrm{d}(p_i)_{i=1}^{\infty})
\end{aligned}$$

We now use the Fubini-Tonelli theorem to break this integral down into $n+1$ iterated integrals using the fact that $\mathrm{SG}X \cong (\mathrm{G}X)^n \times \mathrm{SG}X$, and noting that it is fine to do because we are working with probability (and therefore finite) measures.

$$\begin{aligned}
&\mu^{\mathrm{G}}_{\mathrm{S}X}(\mathbf{G}\kappa_X(\kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty}))(W) \\
&\quad = \int_{(p_i)_{i=1}^{n} \in (\mathrm{G}X)^n} \cdots \int_{p \in \mathrm{SG}X} \prod_{i=1}^{n} p_i(W_i)(\kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty}(\mathrm{d}p_1 \ldots \mathrm{d}p_n \, \mathrm{d}p))
\end{aligned}$$

We can now bring the $\prod_{i=1}^{n} p_i(W_i)$ factor out of the last integral.

$$= \int_{(p_i)_{i=1}^n \in (\mathrm{G}X)^n} \cdots \int \prod_{i=1}^{n} p_i(W_i) \int_{p \in \mathrm{G}X} \kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty}(\mathrm{d}p_1 \ldots \mathrm{d}p_n \, \mathrm{d}p)$$

The innermost integral can now be computed to get $\kappa_{\mathrm{G}X}(\nu_i)_{i=1}^{\infty}(\mathrm{d}p_1 \times \cdots \times \mathrm{d}p_n \times \mathrm{G}X)$, which (by definition of $\kappa_{\mathrm{G}X}$) is just $\prod_{i=1}^{n} \nu_i(\mathrm{d}p_i)$.

$$\begin{aligned}
&= \int_{(p_i)_{i=1}^n \in (\mathrm{G}X)^n} \cdots \int \left( \prod_{i=1}^{n} p_i(W_i) \right) \left( \prod_{i=1}^{n} \nu_i(\mathrm{d}p_i) \right) \\
&= \int_{(p_i)_{i=1}^n \in (\mathrm{G}X)^n} \cdots \int \prod_{i=1}^{n} p_i(W_i) \nu_i(\mathrm{d}p_i) \\
&= \prod_{i=1}^{n} \int_{p_i \in \mathrm{G}X} p_i(W_i) \nu_i(\mathrm{d}p_i)
\end{aligned}$$

And so we are done. It may be easier to see what's really going on in the informal version below. The top path of the pentagon yields the following transformation.

$$(\nu_1, \nu_2, \ldots) \in \mathbf{SGG}X$$

$$\xmapsto{\kappa_{\mathrm{G}X}}$$

$$\{\!\{ (p_1, p_2, \ldots) \mid p_1 \leftarrow \nu_1, p_2 \leftarrow \nu_2, \ldots \}\!\} \in \mathbf{GSG}X$$

$$\xmapsto{\mathbf{G}(\kappa_X)}$$

$$\{\!\{ \kappa_X(p_1, p_2, \ldots) \mid p_1 \leftarrow \nu_1, p_2 \leftarrow \nu_2, \ldots \}\!\} \in \mathbf{GGS}X$$

$$= \{ \text{ definition of } \kappa_X \}$$

$$\{\!\{ \{\!\{ (x_1, x_2, \ldots) \mid x_1 \leftarrow p_1, x_2 \leftarrow p_2, \ldots \}\!\} \mid p_1 \leftarrow \nu_1, p_2 \leftarrow \nu_2, \ldots \}\!\} \in \mathbf{GGS}X$$

$$\xmapsto{\mu^{\mathrm{G}}_{\mathrm{S}X}}$$

$$\{\!\{ (x_1, x_2, \ldots) \mid p_1 \leftarrow \nu_1, p_2 \leftarrow \nu_2, \ldots, x_1 \leftarrow p_1, x_2 \leftarrow p_2, \ldots \}\!\} \in \mathbf{GS}X$$

$$= \{ \text{ commutativity of } \mathbf{G} \}$$

$$\{\!\{ (x_1, x_2, \ldots) \mid p_1 \leftarrow \nu_1, x_1 \leftarrow p_1, p_2 \leftarrow \nu_2, x_2 \leftarrow p_2, \ldots \}\!\} \in \mathbf{GS}X$$

The final two steps are a little imprecise because the application of $\mu^{\mathrm{G}}_{\mathrm{S}X}$ gives us a "monad comprehension" where there are first an infinite number of samples (one from each of $\nu_i$), and then another infinite number (one from each sample of $\nu_i$). We use commutativity of G to rewrite it in a way such that the expression can be read more "productively" (that is, in a way such that any sampling step can be arrived at in a finite number of steps). In our formal proof this was made precise by making use of the Fubini-Tonelli theorem. Now consider the transformations along the bottom path.

$$
\begin{aligned}
&(\nu_1, \nu_2, \dots) \in \mathrm{SGG}X \\
&\qquad \xmapsto{\mathrm{S}(\mu^{\mathrm{G}}_X)} \\
&(\mu^{\mathrm{G}}_X(\nu_1), \mu^{G}_X(\nu_2), \dots) \in \mathrm{SG}X \\
&\qquad \xmapsto{\kappa_X} \\
&\{\!|(x_1, x_2, \dots) \mid x_1 \leftarrow \mu^{\mathrm{G}}_X(\nu_1), x_2 \leftarrow \mu^{\mathrm{G}}_X(\nu_2), \dots|\!\} \in \mathrm{GS}X \\
&\qquad = \{\ \mu^{\mathrm{G}}_X \text{ as a monad comprehension: } x_i \leftarrow \mu^{\mathrm{G}}_X(\nu_i) \ \rightsquigarrow\ p_i \leftarrow \nu_i, x_i \leftarrow p_i \ \} \\
&\{\!|(x_1, x_2, \dots) \mid p_1 \leftarrow \nu_1, x_1 \leftarrow p_1, p_2 \leftarrow \nu_2, x_2 \leftarrow p_2, \dots|\!\} \in \mathrm{GS}X
\end{aligned}
$$

**Pentagon II** Consider a stream of streams of measures $(s_i)_{i=1}^{\infty} \in \mathrm{SSG}X$ where $s_i = (p_i^j)_{j=1}^{\infty} \in \mathrm{SG}X$ and $p_i^j \in \mathrm{G}X$.

$$
\begin{array}{ccccc}
\mathrm{SSG}X & \xrightarrow{\mathrm{S}\kappa_X} & \mathrm{SGS}X & \xrightarrow{\kappa_{\mathrm{S}X}} & \mathrm{GSS}X \\
\Big\downarrow{\scriptstyle \mu^{\mathrm{S}}_{\mathrm{G}X}} & & & & \Big\downarrow{\scriptstyle \mathrm{G}\mu^{\mathrm{S}}_X} \\
\mathrm{SG}X & & \xrightarrow[\kappa_X]{} & & \mathrm{GS}X
\end{array}
$$

We start with the path along the left edge and the bottom.

$$
\begin{aligned}
\kappa_X \left( (\mu^{\mathrm{S}}_{\mathrm{G}X})(s_i)_{i=1}^{\infty} \right)(W) &= \kappa_X (p_i^i)_{i=1}^{\infty}(W) \\
&= \prod_{i=1}^{n} p_i^i(W_i)
\end{aligned}
$$

Now consider the other path. Overall we will need to compute using the push-forward $\mathrm{G}(\mu^{\mathrm{S}}_X)$, which, given a measure $\nu \in \mathrm{GSS}X$, is equal to the measure $W \mapsto \nu((\mu^{\mathrm{S}}_X)^{-1}[W])$. And so let us first explicitly compute what the inverse image of $\mu^{\mathrm{S}}_X$ looks like. Take $W = W_1 \times \cdots \times W_n \times \mathrm{S}X \in \Sigma_{\mathrm{S}X}$ as we have done in all the proofs so far.

$$\begin{aligned}(\mu^{\mathrm{S}}_X)^{-1}[W_1 \times \cdots \times W_n \times \mathrm{S}X] &= \{(s_1, \dots) \mid \mu_X(s_1, \dots) \in W_1 \times W_2 \times \dots\} \\ &= \{(s_1, \dots) \mid (\pi_1(s_1), \pi_2(s_2) \dots) \in W_1 \times W_2 \times \dots\} \\ &= (W_1 \times X \times X \times \cdots) \times (X \times W_2 \times X \times \cdots) \times \cdots\end{aligned}$$

That is, the inverse image is an infinite product of infinite products where all the elements are $X$ except for the diagonals, which are $W_i$. Using this we can compute the entire path.

$$\begin{aligned}(\mathrm{G}\mu^{\mathrm{S}}_X)(\kappa_{\mathrm{S}X}(\mathrm{S}\kappa_X(s_i)_{i=1}^{\infty}))(W) &= (\mathrm{G}\mu^{\mathrm{S}}_X)(\kappa_{\mathrm{S}X}((\kappa_X s_i)_{i=1}^{\infty}))(W) \\ &= \kappa_{\mathrm{S}X}((\kappa_X s_i)_{i=1}^{\infty})\left((\mu^{\mathrm{S}}_X)^{-1}[W]\right)\end{aligned}$$

As we just saw, the inverse image will be an infinite product of products, call it $V_1 \times V_2 \times \dots$, where $V_i = X^{i-1} \times W_i \times \mathrm{S}X$ for $i \leq n$, and $V_i = \mathrm{S}X$ otherwise. Since $V_i$ for $i > n$ is the whole space, we can use the definition of $\kappa_{\mathrm{S}X}$ to simplify this to just a product of the first $n$ terms, $\kappa_X(s_1)(V_1) \times \cdots \times \kappa_X(s_n)(V_n)$. And now again we simplify this expression using the fact that in each $V_i$ all the terms are $X$ except for the $i^{\text{th}}$ term, which is $W_i$. This gives us the final result.

$$(\mathrm{G}\mu^{\mathrm{S}}_X)(\kappa_{\mathrm{S}X}(\mathrm{S}\kappa_X(s_i)_{i=1}^{\infty}))(W) = \prod_{i=1}^{n} p^i_i(W_i)$$

Again, we conclude with the informal monad comprehension version of this proof. The top path of the pentagon yields the following transformation.

$$\begin{aligned}&(\dots,\ s_i,\ \dots) \in \mathrm{SSG}X \\ &\qquad \xmapsto{\mathrm{S}\kappa_X} \\ &(\dots,\ \kappa_X(s_i),\ \dots) \in \mathrm{SGS}X \\ &\qquad = \{\text{ definition of } \kappa_X \} \\ &(\dots,\ \{\!|(\dots,\ x^j_i,\ \dots) \mid \dots,\ x^j_i \leftarrow p^j_i,\ \dots|\!\},\ \dots) \in \mathrm{SGS}X\end{aligned}$$

$$\xmapsto{\kappa_{\mathrm{S}X}}$$

$$\{\!\{(\ldots,\ t_i,\ \ldots) \mid \ldots,\ t_i \leftarrow \{\!\{(\ldots,\ x_i^j,\ \ldots) \mid \ldots,\ x_i^j \leftarrow p_i^j,\ \ldots\}\!\},\ \ldots\ \}\!\} \in \mathrm{GSS}X$$

$$= \{\ \text{by the monad laws}\ \}$$

$$\{\!\{(\ldots,\ (\ldots,\ x_i^j,\ \ldots),\ \ldots) \mid \ldots,\ x_i^j \leftarrow p_i^j,\ \ldots\ \}\!\} \in \mathrm{GSS}X$$

$$\xmapsto{\mathrm{G}(\mu^{\mathrm{S}}_X)}$$

$$\{\!\{\mu^{\mathrm{S}}_X(\ldots,\ (\ldots,\ x_i^j,\ \ldots),\ \ldots) \mid \ldots,\ x_i^j \leftarrow p_i^j,\ \ldots\ \}\!\} \in \mathrm{GS}X$$

$$= \{\ \text{definition of } \mu^{\mathrm{S}}_X\ \}$$

$$\{\!\{(\ldots,\ x_i^i,\ \ldots) \mid \ldots,\ x_i^i \leftarrow p_i^i,\ \ldots\}\!\} \in \mathrm{GS}X$$

Along the bottom path we get the same result.

$$(\ldots,\ s_i,\ \ldots) \in \mathrm{SSG}X$$

$$\xmapsto{\mu^{\mathrm{S}}_{\mathrm{G}X}}$$

$$(\ldots,\ p_i^i,\ \ldots) \in \mathrm{SG}X$$

$$\xmapsto{\kappa_X}$$

$$\{\!\{(\ldots,\ x_i^i,\ \ldots) \mid \ldots,\ x_i^i \leftarrow p_i^i,\ \ldots\}\!\} \in \mathrm{GS}X$$

□

A long-term problem in programming semantics has been the combination of probability and non-determinism. In that context, it is well known that there is no distributive law between the finite distribution monad and the powerset monad (e.g. [Goy and Petrisan, 2020, Varacca and Winskel, 2006, Zwart and Marsden, 2019]), with the first such proof attributed to Plotkin in [Varacca and Winskel, 2006]. In their work, Zwart and Marsden [2019] generalised Plotkin's impossibility result to many other combinations of monads on **Set**. Note that our impossibility result (Theorem 4.5.4) does not arise as a special case of Zwart and Marsden's general results. It is a different kind of argument altogether, and it would be interesting to see how such a technique can be generalised for other monads. Regarding weak distributive laws, in recent work Goy and Petrisan [2020] managed to prove the existence of a weak distributive law between the finite distribution

and powerset monads via an involved construction, making use of results from linear programming.

Our proof of the non-existence of a distributive law between $\mathbf{S}$ and $\mathbf{G}$ is reminiscent of Plotkin's result. Even though we give a direct proof of the weak distributive law, we note that there is scope in presenting it in a cleaner more syntactic manner, but that rests on defining the distributive law via a universal property. This can be done by showing Ionescu-Tulcea's construction to be a categorical limit of some particular diagram in the Kleisli category of the Giry monad, similar to Bochner's presentation of Kolmogorov's extension theorem as a projective limit. From this we may be able to derive general conditions for when weak distributive laws between monads exist. We leave this for future work. It is also worth noting that all the related work mentioned has been within the context of **Set**, whereas we have worked exclusively in **Meas**. Finally, it is unknown whether or not $\mathbf{G} \circ \mathbf{S}$ admits a monad, which we also leave as future research.

## 4.6 Summary and related work

In the set-theoretic case, it *is* possible to find a distributive law between the bag monad and the finite distributions monad (e.g. [Keimel and Plotkin, 2017, App. A], Dahlqvist et al. [2018], Zwart and Marsden [2019]). There is also a result due to Hyland et al. [2006] which states that if $S : C \to C$ is a monad for commutative monads in cocomplete symmetric monoidal closed $C$ and $T : C \to C$ is any arbitrary commutative monad, then there is a distributive law making $T \circ S : C \to C$ a monad. In particular this shows that there is a distributive law between the bag and distribution monads in **Set**. The same result does not immediately extend to **Meas**, however, as it is not monoidal closed. Our construction, where we also make use of the fact that $\mathbf{B}X$ is the free commutative monoid in **Meas** to define the distributive law, is similar to this result without requiring monoidal closedness. Finally, another general result along similar lines is due to Manes and Mulry [2007, Theorem 4.3.4]. It follows from their theorem that the there is a distributive law between $l : \mathbf{BD} \to \mathbf{DB}$ where $\mathbf{B}$ is the bag monad and $\mathbf{D}$ is the finite distributions monad in **Set**. Their approach makes use of algebraic theories and is restricted to **Set**.

# Chapter 5

# The intensity measure

In probability theory the expectation of a probability measure on the reals is the average value of the outcomes, weighted by their respective probabilities. When the number of outcomes is countable, this is given by the sum of products of all the outcomes with their respective probabilities, and when it is uncountable (that is, when the probability density is some function on $\mathbb{R}$ with uncountable support) it is calculated by integrating the outcomes with respect to the probability measure.

This notion extends naturally to the study of point processes where a useful characteristic in describing them is the expected number of points in a given region. This is known as the intensity measure of a point process and is a widely-used concept in point process theory. For example, in the previous chapter in Figure 4.3 we illustrated the homogeneous Poisson process with rate $\Lambda = 10$. The expected number of points in any region is proportional to $10a$ where $a$ is the area of the region. More generally, the *intensity measure* of a point process is the measure that assigns to each measurable subset the expected number points in it. In this chapter we look at compositionally computing the intensity measure of monadic point processes by defining an appropriate monad morphism from the point process monad to the monad of all measures.

**Outline.** We first introduce the all-measures monad $\mathbf{M} : \mathbf{Meas} \to \mathbf{Meas}$ — an analogue of the Giry monad, but for general measures — in Section 5.1. There is a function $\mathbb{E} : \mathbf{GB}X \to \mathbf{M}X$ that takes a point process to its intensity measure (Theorem 5.2.3). We show that this function can be defined naturally by making use of the injections of the Giry and bag monads, which are also monad morphisms

(Theorem 5.2.1), into the all-measures monad. In Section 5.2 we show that this function is actually a monad morphism from the point process monad **GB** to the monad of all measures **M** (Theorem 5.2.5). Thus, if we build a point process using the monadic constructions (for example by composing morphisms in the Kleisli category of **GB**) then we can read off its intensity measure in a compositional way. We show two applications of this monad morphism in Section 5.3. First we use it to calculate the intensity of a Poisson point process, and then we give a new proof of a discrete version of Wald's lemma (a core result in probability theory) as an instance of a specific calculation using our monad morphism.

## 5.1 The monad of all measures

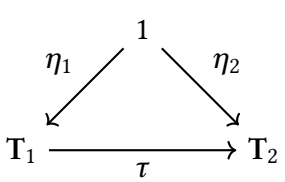
Similar to the Giry monad, we define the monad of all measures on $(X, \Sigma_X)$ by taking the space to be the set of all measures on $X$.

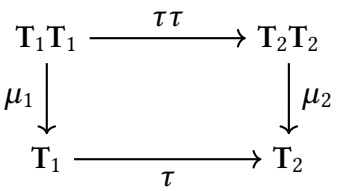
$$\mathbf{M}X = \{\nu : \Sigma_X \to \mathbb{R}^+_\infty \mid \nu \text{ is a measure}\}$$

The $\sigma$-algebra for $\mathbf{M}X$ is given by $\Sigma_{\mathbf{M}X}$, the smallest $\sigma$-algebra such that all the evaluation maps $\mathrm{ev}_U = \nu \mapsto \nu(U) : \mathbf{M}X \to \mathbb{R}^+_\infty$ are measurable (where $U \in \Sigma_X$). Note that unlike the Giry monad, the all-measures monad is *not strong*. The unit and multiplication maps for the all-measures monad are the same as that for the Giry monad, except that the integral as a result of the multiplication need not be bounded by 1 and can be $\infty$. That **M** is not strong comes from the fact that there is no Fubini theorem for arbitrary measures (Fubini's theorem only holds for $\sigma$-finite measures, which is a property that all probability measures have).

Next we define what it means for a natural transformation to be a monad morphism.

**Definition 5.1.1.** Given monads $(\mathbf{T}_1 : C \to C, \eta_1, \mu_1)$ and $(\mathbf{T}_2 : C \to C, \eta_2, \mu_2)$ on $C$, a monad morphism is a natural transformation $\tau_X : \mathbf{T}_1 X \to \mathbf{T}_2 X$ respecting the monad structure. That is, the following diagrams commute.



## 5.2 The intensity measure as a monad morphism

Recall from Section 4.3.2 that for a point process $\gamma \in \mathbf{GB}X$ the probability of observing $k$ points in the region $U \in \Sigma_X$ is given by the expression $p_k = \gamma(\alpha_k^U)$. Then, by definition of expectation, the expected number of points in $U$ is the weighted sum $\sum_{k=0}^{\infty} k \cdot p_k = \sum_{k=0}^{\infty} k \cdot \gamma(\alpha_k^U)$.

In general, the expectation morphism $\mathbb{E}$ maps point processes $\gamma \in \mathbf{GB}X$ to the measure $W \mapsto \sum_{k=0}^{\infty} k \cdot \gamma(\alpha_k^W) : \Sigma_X \to [0, +\infty]$. We first show how to understand this in a more abstract way, by injecting both probability measures $\mathbf{G}X$ and bags $\mathbf{B}X$ each into the all-measures monad $\mathbf{M}X$, in a measurable and natural way.

The measurable inclusion $i_X^{\mathbf{G}} : \mathbf{G}X \hookrightarrow \mathbf{M}X$ is straightforward because $\mathbf{G}X$ is a subset of $\mathbf{M}X$. This trivially makes it a monad morphism from $\mathbf{G}$ to $\mathbf{M}$.

A little more care is needed for the map $i_X^{\mathbf{B}} : \mathbf{B}X \to \mathbf{M}X$. The idea is that we would like to define bags $[\![x_1, \ldots, x_n]\!] \in \mathbf{B}X$ as the sum of Dirac deltas centered around the respective $x_i$ elements – $\sum_{i=1}^{n} \delta_{x_i} \in \mathbf{M}X$. We first note that this is not an injection in general since in the two-point indiscrete measurable space $(\mathbf{2}, \{\varnothing, \mathbf{2}\})$ there are two singleton bags but only one singleton measure; we require our points to be sufficiently separated in order to think of a bag as the sum of Dirac deltas. And so, $i_X^{\mathbf{B}}$ *is injective when* $X$ *is standard Borel. Due to this we assume all measurable spaces in this chapter are standard Borel.*

This injection is familiar in point process theory, indeed many authors actually define $\mathbf{B}X$ as a space of integer-valued measures in the first place.

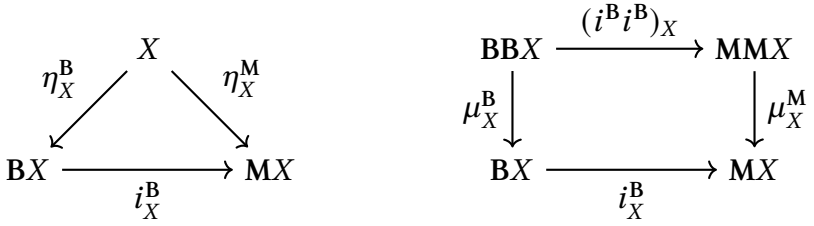
### 5.2.1 Bags as measures

We would like to regard a non-empty bag $[\![x_1, \ldots, x_n]\!] \in \mathbf{B}X$ as the multiplicity-respecting sum of Dirac deltas $\sum_{i=1}^{n} \delta_{x_i}$, and an empty bag $\varnothing \in \mathbf{B}X$ as the degenerate measure $\delta^0 \in \mathbf{M}X$ which sends all measurable sets $W \in \Sigma_X$ to 0. We can define this concisely and measurably using our fold construction from Section 3.4.2.

We start with some bag $b \in \mathbf{B}X$. To this we apply $\mathbf{B}\eta_X^{\mathbf{M}} : \mathbf{B}X \to \mathbf{BM}X$ to get a bag of Dirac deltas, which we can then add by folding over the bag. The function we will be folding with is just $+ = (m_1, m_2) \mapsto (W \in \Sigma_X \mapsto m_1(W) + m_2(W)) : \mathbf{M}X \times \mathbf{M}X \to \mathbf{M}X$ – it is clear that it is measurable and symmetric. For the base case we can just return the degenerate measure $\delta^0 \in \mathbf{M}X$. Combining this into one

definition, we get $i^{\mathbf{B}}_X = b \mapsto \langle + \rangle(\mathbf{B}\eta^{\mathbf{M}}_X(b), \delta^0) : \mathbf{B}X \to \mathbf{M}X$. And so $i^{\mathbf{B}}_X(\varnothing) = \delta^0$ and $i^{\mathbf{B}}_X(\llbracket x_1, \ldots, x_n \rrbracket) = \sum_{i=1}^n \delta_{x_i}$. Next we show that $i^{\mathbf{B}}_X$ is a morphism of monads.

**Theorem 5.2.1.** *$i^{\mathbf{B}}_X : \mathbf{B}X \to \mathbf{M}X$ is a monad morphism, making the following diagrams commute.*




*Proof.* The unit diagram holds since $i^{\mathbf{B}}_X(\eta^{\mathbf{B}}_X(x)) = i^{\mathbf{B}}_X(\llbracket x \rrbracket) = \delta_x = \eta^{\mathbf{M}}_X(x)$. For the monad multiplication diagram it is clear that it holds when we consider $\varnothing \in \mathbf{BB}X$. Now consider $\llbracket b_1, \ldots, b_n \rrbracket \in \mathbf{BB}X$ where $b_i = \llbracket x^i_1, \ldots, x^i_{m_i} \rrbracket \in \mathbf{B}X$. Going down and along the edge we get the nested sum $\sum_{i=1}^n \sum_{j=1}^{m_i} \delta_{x^i_j}$. Now along the top edge we need to apply $(i^{\mathbf{B}}i^{\mathbf{B}})_X : \mathbf{BB}X \to \mathbf{MM}X$, which is shorthand for first applying $\mathbf{B}i^{\mathbf{B}}_X : \mathbf{BB}X \to \mathbf{BM}X$, and then applying $i^{\mathbf{B}}_{\mathbf{M}X} : \mathbf{BM}X \to \mathbf{MM}X$. Applying $\mathbf{B}i^{\mathbf{B}}_X$ gives us $\llbracket i^{\mathbf{B}}_X(b_1), \ldots, i^{\mathbf{B}}_X(b_n) \rrbracket \in \mathbf{BM}X$, to which applying $i^{\mathbf{B}}_{\mathbf{M}X}$ gives us $\sum_{i=1}^n \delta_{i^{\mathbf{B}}_X(b_i)} \in \mathbf{MM}X$. We finally go down the left edge by applying $\mu^{\mathbf{B}}_X$ to this.

$$
\begin{aligned}
\mu^{\mathbf{B}}_X\left(\sum_{i=1}^n \delta_{i^{\mathbf{B}}_X(b_i)}\right) &= U \in \Sigma_X \mapsto \int_{\nu \in \mathbf{M}X} \mathrm{ev}_U(\nu)\ \mathrm{d}\left(\sum_{i=1}^n \delta_{i^{\mathbf{B}}_X(b_i)}(\nu)\right) \\
&= U \in \Sigma_X \mapsto \sum_{i=1}^n \int_{\nu \in \mathbf{M}X} \nu(U)\ \mathrm{d}\delta_{i^{\mathbf{B}}_X(b_i)}(\nu) \\
&= U \in \Sigma_X \mapsto \sum_{i=1}^n \int_{\nu \in \mathbf{M}X} \nu(U)\delta_{i^{\mathbf{B}}_X(b_i)}(\mathrm{d}\nu) \\
&= U \in \Sigma_X \mapsto \sum_{i=1}^n i^{\mathbf{B}}_X(b_i)(U) \\
&= \sum_{i=1}^n i^{\mathbf{B}}_X(b_i) = \sum_{i=1}^n i^{\mathbf{B}}_X(\llbracket x^i_1, \ldots, x^i_{m_i} \rrbracket) \\
&= \sum_{i=1}^n \sum_{j=1}^{m_i} \delta_{x^i_j}
\end{aligned}
$$

□

### 5.2.2 The intensity measure morphism

Having defined $i_X^{\mathbf{B}} : \mathbf{B}X \to \mathbf{M}X$, we use it and $i^{\mathbf{G}} : \mathbf{G}X \to \mathbf{M}X$ to define the intensity measure morphism $\mathbb{E}_X : \mathbf{GB}X \to \mathbf{M}X$.

**Definition 5.2.2.** The intensity measure morphism $\mathbb{E} : \mathbf{GB} \to \mathbf{M}$ is given by first injecting each of the **G** and **B** monads into **M**, after which we use the monad multiplication of **M** to get the required result.

$$\mathbb{E} : \mathbf{GB} \xrightarrow{i^{\mathbf{G}} i^{\mathbf{B}}} \mathbf{MM} \xrightarrow{\mu^{\mathbf{M}}} \mathbf{M}$$

We show that this definition of $\mathbb{E}$ does indeed return the intensity measure of a point process. For it we make use of the change-of-variables formula for integrating measures in order to integrate over pushforward measures.

**Theorem 5.2.3.** *For any point process $\gamma \in \mathbf{GB}X$, $\mathbb{E}(\gamma)(U) = \sum_{k=0}^{\infty} k \cdot \gamma(\alpha_k^U)$.*

*Proof.* Let $\gamma \in \mathbf{GB}X$ and $U \in \Sigma_X$. Then we directly compute $\mathbb{E}(\gamma)(U)$.

$$\begin{aligned}
\mathbb{E}(\gamma)(U) &= \left(\mu^{\mathbf{M}} \circ \mathbf{M}(i_X^{\mathbf{B}}) \circ i_{\mathbf{B}X}^{\mathbf{G}}\right)(\gamma)(U) \\
&= \mu^{\mathbf{M}}(\mathbf{M}(i_X^{\mathbf{B}})(\gamma))(U) \\
&= \int_{\nu \in \mathbf{M}X} \mathrm{ev}_U(\nu)\ (\mathbf{M}(i_X^{\mathbf{B}})(\gamma))(\mathrm{d}\nu) \\
&= \int_{\nu \in \mathbf{M}X} \mathrm{ev}_U(\nu)\ \left(\gamma \circ (i_X^{\mathbf{B}})^{-1}\right)(\mathrm{d}\nu) \\
&= \int_{b \in \mathbf{B}X} (\mathrm{ev}_U \circ i_X^{\mathbf{B}})(b)\ \gamma(\mathrm{d}b) \quad \text{(change of variables; } \nu \in \mathbf{M}X \text{ to } b \in \mathbf{B}X) \\
&= \int_{b \in \mathbf{B}X} i_X^{\mathbf{B}}(b)(U)\ \gamma(\mathrm{d}b)
\end{aligned}$$

We split our domain of integration $\mathbf{B}X = \bigcup_{k=0}^{\infty} \alpha_k^U$ into the disjoint sets $\alpha_k^U$.

$$= \sum_{k=0}^{\infty} \left( \int_{b \in \alpha_k^U} i_X^{\mathbf{B}}(b)(U)\ \gamma(\mathrm{d}b) \right)$$

Consider separately the integral of $i_X^{\mathbf{B}}(b)(U)\ \gamma(\mathrm{d}b)$ over $b \in \alpha_k^U$. Recall that $\alpha_k^U$ is the set of bags with $k$ elements in $U$, and so each bag in this set has at least $k$ elements, allowing us to write $b = [\![x_1, \ldots, x_n]\!]$ where $n \geq k$. Then $i_X^{\mathbf{B}}([\![x_1, \ldots, x_n]\!])(U) = \sum_{i=1}^n \delta_{x_i}(U)$. But by definition of the space $\alpha_k^U$ that we are integrating over, there are only $k$ lots of $x_i$'s which are in $U$. And so $\sum_{i=1}^n \delta_{x_i}(U) = k$ since $\delta_{x_i}(U) = 1$ for $k$ elements and 0 for the rest. We go back to the main integral, which is now much more manageable.

$$\mathbb{E}(\gamma)(U) = \sum_{k=0}^{\infty} \left( \int_{b \in \alpha_k^U} i_X^{\mathbf{B}}(b)(U)\ \gamma(\mathrm{d}b) \right) = \sum_{k=0}^{\infty} \left( k \cdot \int_{b \in \alpha_k^U} \gamma(\mathrm{d}b) \right) = \sum_{k=0}^{\infty} k \cdot \gamma(\alpha_k^U)$$

□

### 5.2.3 Intensity as a monad morphism

Our main result stems from the fact that the distributive law $l : \mathbf{BG} \to \mathbf{GB}$ interacts well with $i^{\mathbf{G}}$ and $i^{\mathbf{B}}$. In particular, the diagram below commutes, which we prove in the next lemma.

$$\begin{array}{ccccc} \mathbf{BG} & & \xrightarrow{\quad l \quad} & & \mathbf{GB} \\ {\scriptstyle i^{\mathbf{B}} i^{\mathbf{G}}} \downarrow & & & & \downarrow {\scriptstyle i^{\mathbf{G}} i^{\mathbf{B}}} \\ \mathbf{M}^2 & \xrightarrow[\mu^{\mathbf{M}}]{} & \mathbf{M} & \xleftarrow[\mu^{\mathbf{M}}]{} & \mathbf{M}^2 \end{array}$$

**Lemma 5.2.4.** $(\mathbb{E} \circ l =)\ \mu^{\mathbf{M}} \circ i^{\mathbf{G}} i^{\mathbf{B}} \circ l = \mu^{\mathbf{M}} \circ i^{\mathbf{B}} i^{\mathbf{G}} : \mathbf{BG} \to \mathbf{M}$.

*Proof.* Consider $[\![\nu_1, \ldots, \nu_n]\!] \in \mathbf{BG}X$ and $U \in \Sigma_X$. Going from $\mathbf{BG}X$ to $\mathbf{M}X$ along the left edge we get

$$[\![\nu_1, \ldots, \nu_n]\!] \xmapsto{i^{\mathbf{B}} i^{\mathbf{G}}} \sum_{i=1}^n \delta_{\nu_i} \xmapsto{\mu^{\mathbf{M}}} \left( W \in \Sigma_X \mapsto \int_{\mathbf{M}X} \mathrm{ev}_W\, \mathrm{d} \left( \sum_{i=0}^n \delta_{\nu_i} \right) \right).$$

Apply $U \in \Sigma_X$ to this expression.

$$\int_{\mathbf{M}X} \mathrm{ev}_U\, \mathrm{d} \left( \sum_{i=0}^n \delta_{\nu_i} \right) = \sum_{i=0}^n \int_{\mathbf{M}X} \mathrm{ev}_U\, \mathrm{d}\delta_{\nu_i} = \sum_{i=0}^n \int_{m \in \mathbf{M}X} \mathrm{ev}_U\, \delta_{\nu_i}(\mathrm{d}m) = \sum_{i=0}^n \nu_i(U)$$

Along the other edge we obtain

$$\begin{aligned}(\mu^{\mathbf{M}} \circ i^{\mathrm{G}} i^{\mathrm{B}})(l[\![\nu_1, \ldots, \nu_n]\!])(U) &= \mathbb{E}(l[\![\nu_1, \ldots, \nu_n]\!])(U) \\ &\qquad \{ \text{ by Theorem 5.2.3 } \} \\ &= \sum_{i=0}^{\infty} i \cdot l[\![\nu_1, \ldots, \nu_n]\!](\alpha_i^U) \\ &\qquad \{ \text{ since } l[\![\ldots]\!](\alpha_i^U) = 0 \text{ for } i > n \} \\ &= \sum_{i=0}^{n} i \cdot l[\![\nu_1, \ldots, \nu_n]\!](\alpha_i^U).\end{aligned}$$

Notice that that $l[\![\nu_1, \ldots, \nu_n]\!](\alpha_k^U)$ is simply the coefficient of $x^k$ in the degree-$n$ polynomial $P(x) = \prod_{i=1}(\nu_i(\overline{U}) + \nu_i(U) \cdot x)$. And so using the technique of generating functions (refer to Wilf [2006] for a good introduction), let $P(x) = \sum_{i=1} l[\![\nu_1, \ldots, \nu_n]\!](\alpha_i^U) \cdot x^i$. Differentiating the former expression of $P(x)$ with respect to $x$ gives us

$$P'(x) = \sum_{i=1}^{n} \nu_i(U) \cdot \prod_{j \neq i}^{n} \left(\nu_i(\overline{U}) + \nu_i(U) \cdot x\right).$$

Setting $x = 1$ above, we get $P'(1) = \sum_{i=1}^{n} \nu_i(U) \times 1$ since $\nu_i(\overline{U}) + \nu_i(U) = 1$. Differentiating the latter expression of $P(x)$ we get

$$P'(x) = \sum_{i=1}^{n} i \cdot l[\![\nu_1, \ldots, \nu_n]\!](\alpha_i^U) \cdot x^{i-1}.$$

Setting $x = 1$ here gives us $P'(1) = \sum_{i=1}^{n} i \cdot l[\![\nu_1, \ldots, \nu_n]\!](\alpha_i^U)$. And so the diagram commutes. □

With these results we show that $\mathbb{E}$ is a monad morphism.

**Theorem 5.2.5.** *The intensity measure $\mathbb{E} : \mathbf{GB} \to \mathbf{M}$ is a monad morphism.*

*Proof.* A simple calculation shows the unit law to hold. For the monad multiplication consider the two diagrams in Figure 5.1. All the sub-diagrams in it commute: (I) due to Lemma 5.2.4, (II) by naturality, (III) due to associativity of $\mu^{\mathbf{M}}$, (IV) and (VII) due to $i^{\mathrm{G}}$ and $i^{\mathrm{B}}$ being monad morphisms (Lemma 5.2.1), and finally (V) and (VI)

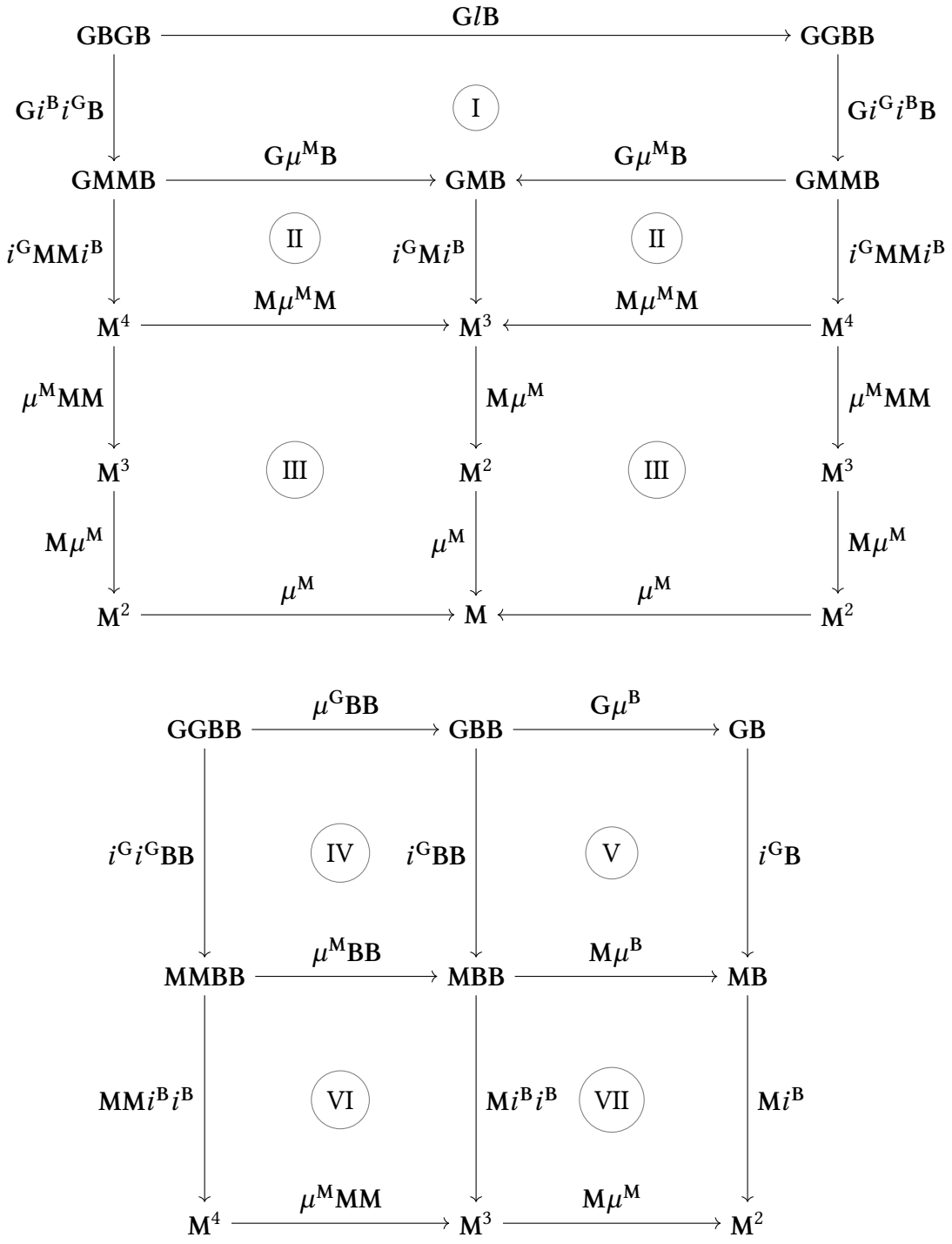



Figure 5.1: Intensity as a monad morphism.

by naturality. Using these commutative diagrams we prove the required equality.

$$
\begin{aligned}
&\mu^{\mathbf{M}} \circ \mathbb{E}\mathbb{E} : \mathbf{GBGB} \to \mathbf{M} \\
&\mu^{\mathbf{M}} \circ \mathbb{E}\mathbb{E} \\
&\qquad \{ \text{ definition of } \mathbb{E} + \text{ naturality } \} \\
&= \mu^{\mathbf{M}} \circ \mathbf{M}\mu^{\mathbf{M}} \circ \mu^{\mathbf{M}}\mathbf{MM} \circ i^{\mathbf{G}}\mathbf{MM}i^{\mathbf{B}} \circ \mathbf{G}i^{\mathbf{B}}i^{\mathbf{G}}\mathbf{B} \\
&\qquad \{ \text{ top diagram (Figure 5.1) } \} \\
&= \mu^{\mathbf{M}} \circ \mathbf{M}\mu^{\mathbf{M}} \circ \mu^{\mathbf{M}}\mathbf{MM} \circ i^{\mathbf{G}}\mathbf{MM}i^{\mathbf{B}} \circ \mathbf{G}i^{\mathbf{G}}i^{\mathbf{B}}\mathbf{B} \circ \mathbf{G}l\mathbf{B} \\
&\qquad \{ \text{ naturality } \} \\
&= \mu^{\mathbf{M}} \circ \mathbf{M}\mu^{\mathbf{M}} \circ \mu^{\mathbf{M}}\mathbf{MM} \circ \mathbf{MM}i^{\mathbf{B}}i^{\mathbf{B}} \circ i^{\mathbf{G}}i^{\mathbf{G}}\mathbf{BB} \circ \mathbf{G}l\mathbf{B} \\
&\qquad \{ \text{ bottom diagram (Figure 5.1) } \} \\
&= \mu^{\mathbf{M}} \circ \mathbf{M}i^{\mathbf{B}} \circ i^{\mathbf{G}}\mathbf{B} \circ \mathbf{G}\mu^{\mathbf{B}} \circ \mu^{\mathbf{G}}\mathbf{BB} \circ \mathbf{G}l\mathbf{B} \\
&\qquad \{ \text{ definition of } \mu^{\mathbf{GB}} \} \\
&= \mu^{\mathbf{M}} \circ \mathbf{M}i^{\mathbf{B}} \circ i^{\mathbf{G}}\mathbf{B} \circ \mu^{\mathbf{GB}} \\
&\qquad \{ \text{ definition of } \mathbb{E} \} \\
&= \mathbb{E} \circ \mu^{\mathbf{GB}}
\end{aligned}
$$

□

## 5.3 Applications of the monad morphism

### 5.3.1 Intensity of a Poisson point process

In Section 4.4.3 we simulated a Poisson point process with rate $\Lambda$ on the unit square by composing the Poisson distribution with a uniform singleton. Here we show this has the required intensity measure ($W \in \Sigma_{[0,1]^2} \mapsto |W| \times \Lambda$, where $|\cdot|$ is the Lebesgue measure) in a compositional way, which we derive using the monad morphism. Let $\underline{N}_\Lambda \in \mathbf{GB1} \cong \mathbf{G}\mathbb{N}$ be the Poisson distribution with rate $\Lambda$, and let $U_2 \in \mathbf{GB}[0,1]^2$ be the uniformly distributed single-point point process. Then recall that the Poisson point process, in monad comprehension notation, is $\pi_\Lambda = \{\!|(x,y) \mid \star \leftarrow \underline{N}_\Lambda, (x,y) \leftarrow U_2|\!\}_{\mathbf{GB}}$. We first de-sugar this to the equivalent expression using the bind $\gg\!=_{\mathbf{GB}}$ and perform some quick simplifications using the

monad laws, after which we end up with the following.

$$\begin{aligned}\pi_\Lambda &= \{\!|(x,y) \mid \star \leftarrow \underline{N}_\Lambda, (x,y) \leftarrow U_2|\!\}_{\mathbf{GB}} \\ &= \underline{N}_\Lambda \gg\!=_{\mathbf{GB}} (\star \mapsto U_2)\end{aligned}$$

Now we apply $\mathbb{E}$. Because it is a monad morphism, it distributes over the bind of **GB**, also converting the bind to that of **M**. After this we proceed with just computing an integral, which we spell out step-by-step.

$$\begin{aligned}\mathbb{E}(\pi_\Lambda) &= \mathbb{E}\left(\underline{N}_\Lambda \gg\!=_{\mathbf{GB}} (\star \mapsto U_2)\right) \\ &= \mathbb{E}\underline{N}_\Lambda \gg\!=_{\mathbf{M}} (\star \mapsto \mathbb{E}U_2) \\ &= \mu^{\mathbf{M}}\left(\mathbf{M}(\star \mapsto \mathbb{E}U_2)(\mathbb{E}\underline{N}_\Lambda)\right) \\ &= W \in \Sigma_{[0,1]^2} \mapsto \int_{\mathbf{M}[0,1]^2} \mathrm{ev}_W \,\mathrm{d}\left(\mathbf{M}(\star \mapsto \mathbb{E}U_2)\left(\mathbb{E}\underline{N}_\Lambda\right)\right) \\ &\qquad \{\text{ change of variables }\} \\ &= W \in \Sigma_{[0,1]^2} \mapsto \int_{\mathbf{1}} \mathrm{ev}_W \circ (\star \mapsto \mathbb{E}U_2)\,\mathrm{d}\left(\mathbb{E}\underline{N}_\Lambda\right) \\ &\qquad \{\text{ introducing explicit variables }\} \\ &= W \in \Sigma_{[0,1]^2} \mapsto \int_{s\in\mathbf{1}} (\mathrm{ev}_W \circ (\star \mapsto \mathbb{E}U_2))(s)\ \mathbb{E}\underline{N}_\Lambda(\mathrm{d}s) \\ &\qquad \{\ \mathbf{1} = \{\star\} \text{ and so the integral reduces to a single term }\} \\ &= W \in \Sigma_{[0,1]^2} \mapsto (\mathrm{ev}_W \circ (\star \mapsto \mathbb{E}U_2))(\star) \times \mathbb{E}\underline{N}_\Lambda(\{\star\}) \\ &= W \in \Sigma_{[0,1]^2} \mapsto \mathrm{ev}_W(\mathbb{E}U_2) \times \mathbb{E}\underline{N}_\Lambda(\{\star\}) \\ &= W \in \Sigma_{[0,1]^2} \mapsto \mathbb{E}U_2(W) \times \mathbb{E}\underline{N}_\Lambda(\{\star\}) \\ &\qquad \{\text{ by definitions of the expectations of } U_2 \text{ and } \underline{N}_\Lambda\ \} \\ &= W \in \Sigma_{[0,1]^2} \mapsto |W| \times \Lambda\end{aligned}$$

And so we have arrived at the correct intensity measure of the Poisson point process, $\mathbb{E}(\pi_\Lambda)(W) = |W| \times \Lambda$.

### 5.3.2 Discrete Wald's lemma

Let $(X_i)_{i\in\mathbb{N}}$ be a sequence of i.i.d. real-valued random variables and let $N$ be an integer-valued random variable, and suppose both have finite expectations. There is a well-known result in probability theory, known as Wald's lemma [Wald, 1944],

which relates the expectation of the random sum up to $N$ of the i.i.d. $X_i$'s with the expectations of $N$ and $X_i$. In particular, it says that the expectation of the random sum is the product of the expectations of $N$ and $X_i$.

$$\mathbb{E}\left[\sum_{i=1}^{N} X_i\right] = \mathbb{E}[N] \times \mathbb{E}[X_i]$$

Using our monad morphism we can derive this identity in the discrete case, i.e. when the $X_i$'s are integer-valued random variables.

Recall our construction of compound probability distributions using the point process monad from Section 4.4.1. In it we regarded $\underline{N}$ and $\underline{X} \in \mathbf{GB1}$ as probability distributions on the naturals (via the isomorphism $\mathbb{N} \cong \mathbf{B1}$) and composed them using our monad.

$$\begin{aligned}
&\gamma \in \mathbf{GB1} \\
&\gamma = \{\!|\star \mid \star \leftarrow \underline{N}, \star \leftarrow \underline{X}|\!\}_{\mathbf{GB}}
\end{aligned}$$

The intensity measure of this point process (applied to $\mathbf{1}$, which is the whole space this point process is defined on) is exactly the expectation of the random sum. As in the previous section, we first de-sugar the monad comprehension to obtain a simplified bind expression.

$$\gamma = \underline{N}_{\Lambda} \gg\!=_{\mathbf{GB}} (\star \mapsto \underline{X})$$

We now apply $\mathbb{E}$ to this expression and have the $\mathbb{E}$ distribute over $\gg\!=_{\mathbf{GB}}$, obtaining a similar expression.

$$\mathbb{E}(\gamma) = \mathbb{E}(\underline{N}_{\Lambda}) \gg\!=_{\mathbf{M}} (\star \mapsto \mathbb{E}(\underline{X}))$$

Upon computing the same integral as before, we end up with the expected expected value, which is the product of expectations.

$$\mathbb{E}(\gamma) = W \in \Sigma_{\mathbf{1}} \mapsto \mathbb{E}(\underline{N}_{\Lambda})(W) \times \mathbb{E}(\underline{X})(W)$$

There is only one non-empty $W$ in $\Sigma_{\mathbf{1}}$, which is $\mathbf{1} = \{\star\}$ itself. And so the result is isomorphic to $\mathbb{E}(\underline{N}_{\Lambda})(\mathbf{1}) \times \mathbb{E}(\underline{X})(\mathbf{1})$, which is indeed $\mathbb{E}[N] \times \mathbb{E}[X_i]$, proving Wald's lemma. With this we have shown a discrete version of Wald's lemma to fall out of

our monad morphism as a special case.

## 5.4 Related work

We remark that a natural transformation in the *opposite* direction ($\mathbf{M}^+ \to \mathbf{GB}$) has been exhibited by Dahlqvist et al. [2016], where $\mathbf{M}^+X$ is the space of finite non-empty measures. This natural transformation takes an intensity measure to the corresponding inhomogeneous Poisson process. Since $\mathbf{M}^+$ is not a monad, it remains to be seen whether this natural transformation can be made into a monad morphism somehow.

# Chapter 6

# Probabilistic databases as finite point processes

Probabilistic databases cater for uncertainty in data. There may be uncertainty about whether rows should be in a database, or uncertainty about what values certain attributes should have. In this final chapter we give an application of our point process monad to show how probabilistic databases can be regarded as finite point processes and queries on them as measurable morphisms between point processes.

Let us motivate a real-world scenario where a probabilistic database can be used to model data, borrowing an example from Grohe et al. [2020, Example 1.2]. Consider a particle detector of some sort, from which we record the time, position, and velocities of detected particles in a relation of schema $\mathsf{Time} \times \mathsf{Position} \times \mathsf{Velocity}$. The individual measurements of position and velocity are always noisy and are best modelled by a probability distribution. But there is an additional source of noise here, and that is in the number of particles detected – some particles may go undetected, and also some false-positives may show up in the data. And so the number of tuples in this database will also be random. What is guaranteed, however, is that there will always only be a finite amount of tuples. This is exactly a finite point process on the product space $\mathsf{Time} \times \mathsf{Position} \times \mathsf{Velocity}$.

And so, a probabilistic database on a schema $X_1 \times \cdots \times X_n$ should be regarded as a finite point process in $\mathbf{GB}(X_1 \times \cdots \times X_n)$. A good semantic analysis is important in view of the recent work of Grohe and Lindner [2019, 2020] which builds on [Ceylan

et al., 2016, Jampani et al., 2008]. This new line of work breaks with the traditional approach of having a fixed finite support for probabilistic databases, and argues that the support should be infinite, possibly uncountable. This leads to semantic complications and introduces issues of measurability, which we can easily address using our point process monad.

In the deterministic setting, queries translate databases from one schema to another. In particular, a query can be seen as a transformation between bags $\mathbf{B}(X_1 \times \cdots \times X_m) \to \mathbf{B}(Y_1 \times \cdots \times Y_n)$. For probabilistic databases, the usual approach is to consider queries on deterministic databases, and then lift them to probabilistic databases. Semantically, this can regarded as the functorial action of the monad $\mathbf{G}$, which then gives a translation between probabilistic databases. Building on the approach of Grohe and Lindner [2019], we look at queries on probabilistic databases as measurable operations on bags lifted to the probabilistic setting. To this end, we consider an expressive query language on bags (called BALG) and show that it extends to measurable spaces of bags.

*In this chapter we work with the assumption of all our underlying measurable spaces being standard Borel, which is required in order to have a measurable equality predicate $X \times X \to$ Bool. Moreover, some non-standard Borel spaces do not work well with probabilistic database operations, as shown by Grohe and Lindner [2020, Example 8].*

**Outline.** We address two important aspects of probabilistic databases in this chapter. In Section 6.1 we introduce BALG, the query language for bag databases using which we write queries on probabilistic databases. We prove that all BALG queries extend naturally to the measurable setting by giving them a semantics using the monad and monoid on measurable spaces of bags (Theorem 6.1.1). Next in Section 6.1.2 we extend the BALG query language to include a grouping construct, which we also show to be measurable using our bag monad. Finally we touch upon the topic of generating probabilistic databases in Section 6.2 in which we make use of point process constructions we showed to be measurable in Chapter 4.

## 6.1 An algebraic query language for bag databases

In the standard theory of database modelling, relations are assumed to be sets, disallowing the existence of duplicates. Most database software, however, relax this restriction, often to save the cost of duplicate elimination. BALG ("bag algebra"), an

algebra for manipulating bags, was first introduced by Grumbach et al. [1996]. The authors presented BALG as an extension of the nested relation algebra RALG [Klug, 1982, Özsoyoğlu et al., 1987], with a focus on the study of its expressive power and relative complexity to RALG, showing that BALG as a query language was more expressive than RALG.

Previously only a fragment of BALG was shown to be measurable [Grohe and Lindner, 2019], with each language construct requiring a different proof. In this section we show that *entire* BALG query language extends to measurable functions on bags in a uniform way using monad comprehensions. We first briefly review the queries of BALG, after which we give the semantics of these queries as measurable bag functions.

- The SINGLETON operation returns a singleton bag consisting of the input.
- Restructuring rows of tables is possible using the $\text{MAP}_f$ query, which applies the function $f$ to every row in the table.
- The queries PROD, DUNION, DIFFERENCE, UNION, and INTERSECT compute the product, disjoint union, difference, union, and intersection of the input tables respectively.
- The PROJECT query projects out user-specified columns.
- FLATTEN transforms a bag of bags to a bag consisting of the disjoint unions of all the internal bags.
- Duplicate elimination, or deduplication, is possible using the DEDUP query.
- Finally, we can compute the bag of sub-bags of any bag using POWERBAG, and the bag of subsets of any bag using POWERSET.

Given the expressiveness of the BALG query language it comes as no surprise that many operations can be defined in terms of each other. For example, the powerset of a bag is simply the deduplicated version of the powerbag. It is also known that the union and intersection of bags can be defined using the disjoint union and difference operators. To this end we will only consider the following minimal subset of BALG queries, in terms of which all other queries can be defined: {SINGLETON, FLATTEN, MAP, PRODUCT, PROJECT, SELECT, DUNION, DIFFERENCE, POWERBAG, DEDUP}.

### 6.1.1 Measurability of BALG queries

We provide a semantics to BALG queries by mapping each query to a measurable function on bags. The measurability of the semantics of the SINGLETON, FLATTEN, MAP, PRODUCT, PROJECT, and SELECT queries is guaranteed by defining their semantics as monad comprehensions. The measurability of the semantics of the remaining queries, DUNION, DIFFERENCE, POWERBAG, and DEDUP, is obtained by defining their semantics using our fold construction introduced in Section 3.6. Note that commutativity holds for all the measurable accumulating functions in the fold-based definitions to follow. The condition of commutativity is easy to check.

**Bagging and flattening.** The semantics for the SINGLETON and FLATTEN queries are given by the unit $\eta^{\mathbf{B}}$ and multiplication $\mu^{\mathbf{B}}$ maps for the bag monad $\mathbf{B}$. The measurability of these maps is proved in Theorem 3.6.6.

$$\begin{aligned} &[\![\textsc{singleton}]\!] : X \to \mathbf{B}X && [\![\textsc{flatten}]\!] : \mathbf{B}^2X \to \mathbf{B}X \\ &[\![\textsc{singleton}]\!](x) = \eta^{\mathbf{B}}_X(x) = [\![x]\!] && [\![\textsc{flatten}]\!](b) = \mu^{\mathbf{B}}_X(b) \end{aligned}$$

**Restructuring.** The $\textsc{map}_f$ query comes parametrised with a measurable function $f : X \to Y$ and its semantics is simply the functorial action of $\mathbf{B}$ on $f$, which yields yet another measurable map. This can equivalently be written using monad comprehension notation.

$$\begin{aligned} &[\![\textsc{map}_f]\!] : \mathbf{B}X \to \mathbf{B}Y \\ &[\![\textsc{map}_f]\!](b) = \mathbf{B}(f)(b) = \{\!|\, f(x) \mid x \leftarrow b \,|\!\}_{\mathbf{B}} \end{aligned}$$

**Product and projection.** Monad comprehensions make it straightforward to define the product of two bags where the arity of the resultant schema is the sum of the arities of the input schemas.

$$\begin{aligned} &[\![\textsc{prod}]\!] : \mathbf{B}(X_1 \times \cdots \times X_m) \times \mathbf{B}(Y_1 \times \cdots \times Y_n) \to \mathbf{B}(X_1 \times \cdots \times X_m \times Y_1 \times \cdots \times Y_n) \\ &[\![\textsc{prod}]\!](b_1, b_2) = \{\!|\, (x_1, \ldots, x_m, y_1, \ldots, y_n) \mid (x_1, \ldots, x_m) \leftarrow b_1, \\ &\qquad\qquad\qquad\qquad\qquad\qquad (y_1, \ldots, y_n) \leftarrow b_2 \,|\!\}_{\mathbf{B}} \end{aligned}$$

A similar treatment can be given to the $\textsc{project}_{i_1,\ldots,i_k}$ query which projects out the $i_1, \ldots, i_k$ indices of the input schema.

$$[\![\textsc{project}_{i_1,\ldots,i_k}]\!] : \mathbf{B}(X_1 \times \cdots \times X_n) \to \mathbf{B}(X_{i_1} \times \cdots \times X_{i_k})$$
$$[\![\textsc{project}_{i_1,\ldots,i_k}]\!](b) = \{\!|\, (x_{i_1}, \ldots, x_{i_k}) \mid (x_1, \ldots, x_n) \leftarrow b \,|\!\}_{\mathbf{B}}$$

**Selection.** $\textsc{select}_\psi$ is parametrised by a measurable Boolean predicate $\psi : X \to$ Bool and filters out the rows in the input table satisfying $\psi$. We have previously defined the measurable function $\mathsf{filter}_\psi : \mathbf{B}X \to \mathbf{B}X$ parameterised by measurable $\psi : X \to$ Bool in Section 4.4.3, which we defined measurably a monad comprehension. Recall that below $\hat{\psi} : X \to \mathbf{1}$ is a lifting of $\psi$ – more specifically, it is the composition of $\psi$ with the function which sends True to $[\![\star]\!]$ and False to $\varnothing$.

$$[\![\textsc{select}_\psi]\!] : \mathbf{B}X \to \mathbf{B}X$$
$$[\![\textsc{select}_\psi]\!](b) = \mathsf{filter}_\psi(b) = \{\!|\, x \mid x \leftarrow b,\ \star \leftarrow \hat{\psi}(x) \,|\!\}_{\mathbf{B}}$$

Whenever the predicate $\psi$ evaluates to false, its lifted version $\hat{\psi}$ evaluates to the empty bag, which results in the particular entry being excluded from the resultant bag.

**Disjoint union.** DUNION simply computes the disjoint union of its arguments. In Section 3.6 we defined the measurable disjoint union $\uplus : \mathbf{B}X \times \mathbf{B}X \to \mathbf{B}X$ as the fold of $\mathsf{add} : X \times \mathbf{B}X \to \mathbf{B}X$.

$$[\![\textsc{dunion}]\!] : \mathbf{B}X \times \mathbf{B}X \to \mathbf{B}X$$
$$[\![\textsc{dunion}]\!](b_1, b_2) = b_1 \uplus b_2 = \langle \mathsf{add} \rangle (b_1, b_2)$$

**Bag difference.** Bag difference is the first operation we consider that is not quite immediate from the monad and monoid structure. The idea is that, for instance, $[\![\textsc{difference}]\!]([\![1, 1, 2]\!], [\![1, 2, 3]\!]) = [\![1]\!]$.

Defining bag difference as a fold requires a little care. To this end, we first define the measurable function remove which takes an element and a bag as input and returns either the same bag if that element did not belong to the bag, or a modified bag with one fewer instance of the given element. remove is defined as a fold over the accumulating function $\mathsf{remAcc}_{x_{\mathsf{rem}}} : X \times (\mathsf{Bool} \times \mathbf{B}X) \to \mathsf{Bool} \times \mathbf{B}X$, which is

indexed by the value $x_{\mathsf{rem}} \in X$ it needs to remove from the bag.

The accumulator here is a pair consisting of a boolean flag and the bag into which elements are collected one-by-one. This is so that as we walk through the input bag and keep collecting elements from it into the accumulating bag. The first time we see an $x$ in the input bag that is equal to $x_{\mathsf{rem}}$ we don't add it, and then flip the boolean flag to indicate the removal has been completed. After that we can proceed with adding the rest of the elements to the final bag. We set the convention that the flag False indicates the element has not yet been removed. We thus define remAcc as follows.

$$\mathsf{remAcc}_{x_{\mathsf{rem}}} : X \times (\mathsf{Bool} \times \mathbf{B}X) \to \mathsf{Bool} \times \mathbf{B}X$$

$$\mathsf{remAcc}_{x_{\mathsf{rem}}}(x, (\mathsf{flag}, b)) = \begin{cases} (\mathsf{True}, \mathsf{add}(x, b)) & \text{if flag = True} \\ \begin{cases} (\mathsf{True}, b) & \text{if } x = x_{\mathsf{rem}} \\ (\mathsf{False}, \mathsf{add}(x, b)) & \text{if } x \neq x_{\mathsf{rem}} \end{cases} & \text{if flag = False} \end{cases}$$

Each of the three cases has been defined by combining tupling and add, both of which are measurable functions. The case split on flag is measurably valid as it can be seen as a function coming out of $\mathsf{Bool} = \mathbf{1} + \mathbf{1}$, and the case split on the equality of $x$ and $x_{x_{\mathsf{rem}}}$ is fine due to $X$ being a standard Borel space, and so the diagonal $\{(x, x) \mid x \in X\}$ is a measurable set. From this it follows that $\mathsf{remAcc}_{x_{\mathsf{add}}}$ is measurable. It is also clear from how remAcc was defined that it is a symmetric function. And so we define $\mathsf{remove} : X \times \mathbf{B}X \to \mathbf{B}X$ by folding over remAcc with $(\mathsf{False}, \varnothing)$ as the initial accumulator. The bag we need is then the second projection of the result.

$$\begin{aligned} &\mathsf{remove} : X \times \mathbf{B}X \to \mathbf{B}X \\ &\mathsf{remove}(x_{\mathsf{rem}}, b) = \pi_2 \left( \langle \mathsf{remAcc}_{x_{\mathsf{rem}}} \rangle ((\mathsf{False}, \varnothing), b) \right) \end{aligned}$$

Using remove we define the bag difference of $b_1$ and $b_2$ by letting $b_1$ be the initial input and from it remove-ing each element in $b_2$ one-by-one. The measurability of bag difference follows from the symmetry and measurability of remove.

$$\begin{aligned} &[\![ \textsc{difference} ]\!] : \mathbf{B}X \times \mathbf{B}X \to \mathbf{B}X \\ &[\![ \textsc{difference} ]\!](b_1, b_2) = \langle \mathsf{remove} \rangle (b_1, b_2) \end{aligned}$$

**Powerbag.** The powerbag of a bag is analogous to the powerset of a set, while also respecting multiplicities. As an example, the powerbag of $[\![1, 1]\!]$ is the bag containing the four sub-bags $[\![\varnothing, [\![1]\!], [\![1]\!], [\![1, 1]\!]]\!]$. We measurably define the powerbag of a bag by folding over the accumulating function powerAcc where for every new element $x$ added to the accumulating bag of bags $b_0$ we first add $x$ to every bag in $b_0$ and then take the disjoint union with the initial $b_0$. We define $\mathsf{add}_x$ to be the $x$-section of add (that is, $\mathsf{add}_x(b) = \mathsf{add}(x, b)$). Sections of measurable functions are measurable.

$$\mathsf{powerAcc} : X \times \mathbf{B}^2 X \to \mathbf{B}^2 X \qquad [\![\textsc{powerbag}]\!] : \mathbf{B}X \to \mathbf{B}^2 X$$
$$\mathsf{powerAcc}(x, b_0) = b_0 \uplus \mathbf{B}(\mathsf{add}_x)(b_0) \qquad [\![\textsc{powerbag}]\!](b) = \langle \mathsf{powerAcc} \rangle (\{\![\varnothing]\!\}, b)$$

**Deduplication.** In order to deduplicate a bag (i.e. remove duplicates so that all elements have multiplicity 1) we recurse over its elements and add them to an accumulating bag one-by-one. We avoid multiplicities greater than one in the final bag by first filtering out the value we want to add from the accumulating bag before adding it, which is reflected in the definition of dedupAcc.

$$\mathsf{dedupAcc} : X \times \mathbf{B}X \to \mathbf{B}X \qquad [\![\textsc{dedup}]\!] : \mathbf{B}X \to \mathbf{B}X$$
$$\mathsf{dedupAcc}(x, b) = \mathsf{add}(x, \mathsf{filter}_{\neq x}(b)) \qquad [\![\textsc{dedup}]\!](b) = \langle \mathsf{dedupAcc} \rangle (b, \varnothing)$$

The function $\neq x : X \to \mathsf{Bool}$ is measurable since $\neq x^{-1}(\{\mathsf{True}\}) = X \setminus \{x\}$ and $\neq x^{-1}(\{\mathsf{False}\}) = \{x\}$, both of which are measurable sets (due to $X$ being standard Borel).

**Theorem 6.1.1.** *Balg queries yield measurable functions on bags.*

### 6.1.2 Grouping and aggregation

A standard query in databases is that of grouping and aggregation. Given a database with schema $X \times Y$, the result of a grouping query on it is a table with schema $X \times \mathbf{B}Y$, where the elements of $X$ are paired with the bag of elements they were related to in the given database. Here we introduce a group query to Balg and show that it is a measurable operation on bags.

**Definition 6.1.2.** The $\textsc{group}_{p_1,p_2}$ query acts on tables of schema $X_1 \times \cdots \times X_k$ and is parametrised by two projection functions $p_1 : X_1 \times \cdots \times X_k \to X_{i_1} \times \cdots \times X_{i_m}$

and $p_2 : X_1 \times \cdots \times X_k \to X_{j_1} \times \cdots \times X_{j_n}$. The result of this query is a table with schema $(X_{i_1} \times \ldots X_{i_m}) \times \mathbf{B}(X_{j_1} \times \cdots \times X_{j_n})$ where the elements of the first column are paired with the bag of elements they were related to in the input table. In other words, we group the rows of the table by the elements in the $p_1$-projection of the table.

The measurable bag-semantics for $\textsc{group}_{p_1,p_2}$ is given by

$$[\![\textsc{group}_{p_1,p_2}]\!] : \mathbf{B}(X_1 \times \cdots \times X_k) \to \mathbf{B}((X_{i_1} \times \ldots X_{i_m}) \times \mathbf{B}(X_{j_1} \times \cdots \times X_{j_n}))$$
$$[\![\textsc{group}_{p_1,p_2}]\!](b) = \{\!|\, (i, [\![\textsc{map}_{p_2}]\!](\mathsf{filter}_{\lambda x.p_1(x)\equiv i}(b))) \mid i \leftarrow [\![\textsc{dedup}]\!]([\![\textsc{map}_{p_1}(b)]\!]) \,|\!\}$$

In the monad comprehension we first project out the columns of interest using $p_1$ and deduplicate the resultant bag. From this bag we extract out the rows indices by which we index the rows of the input bag. For each index $i$ we return the pair consisting of $i$ along with the $p_2$-projection of the input where the $p_1$-projection of the rows is equal to $i$. We can conclude that this query is measurable by defining it as a monad comprehension composed of other measurable functions.

Often such grouping queries are followed by aggregations on attributes, the elements of which have been grouped. Examples of aggregation include counting the number of elements in bags, or performing mathematical operations on them such as summing and averaging. As long as we can define such an aggregation function as a measurable function $f$, we can apply a $\textsc{map}$ query to the resultant grouped table. Going back to the example of a probabilistic database with schema $X \times Y$, we can first get a table where the elements in $X$ are related to bags of $Y$ using the $\textsc{group}_{\pi_1,\pi_2}$, to which we can apply an aggregation function $f : \mathbf{B}Y \to Z$ using $\textsc{map}_{X\times f}$.

A second option for defining an group/aggregation query comes from an extension to monad comprehensions in Haskell where the syntax has been extended with the keywords `group by` Peyton Jones and Wadler [2007]. This extension works for any strong monad, but the user needs to provide a grouping function which, in our case, needs to map a bag on $X$ to a bag of bags on $X$ where each sub-bag contains the same element. This can be written in Balg as

$$\textsc{group'}(b) = \textsc{map}_{\lambda i.\textsc{select}_{\lambda x.x\equiv i}(b)}(\textsc{dedup}(b))$$

The measurable bag semantics of this new GROUP' query is then given by the following comprehension.

$$[\![\text{GROUP'}]\!] : \mathbf{B}X \to \mathbf{B}^2 X$$
$$[\![\text{GROUP'}]\!](b) = \{\!\{ \{\!\{ x \mid x \leftarrow b, x = i \}\!\}_{\mathbf{B}} \mid i \leftarrow \text{DEDUP}(b) \}\!\}_{\mathbf{B}}$$

The entire grouping and aggregation query of our toy example probabilistic database $\text{pdb} \in \mathbf{GB}(X \times Y)$ can now be concisely written in the notation of Peyton Jones and Wadler [2007] as

$$\{\!\{ (\text{the } x, f(y)) \mid (x, y) \leftarrow \text{pdb}, \texttt{ group by } x \}\!\}.$$

This modified comprehension syntax works by implicitly changing the types of $x$ and $y$ from $X$ and $Y$ in the right half of the comprehension to $x \in \mathbf{B}X$ and $y \in \mathbf{B}Y$ on the left half. This allows us to apply the measurable aggregation function $f : \mathbf{B}Y \to Z$ to $y$. The aggregation function used on $x$ is $\text{the} : \mathbf{B}X \to X$, which is some measurable function such that $\text{the}(b) = x$ when $b$ is a bag that only contains copies of $x$. (For example, we could sort $b$ and return the first element.)

## 6.2 Generating probabilistic databases

Having looked at measurable queries on probabilistic databases, we now turn to the question of where probabilistic databases come from in the first place, particularly in the setting where they have infinite support. A good language for generating infinite probabilistic databases remains a topic of active research, but we now illustrate that the monads for probability **G** and bags **B** could be the basis of a good intermediate language.

### 6.2.1 Randomising attributes

For a first example of a probabilistic database, suppose we are given a deterministic bag of $\text{Time} \times \text{Velocity}$ pairs from the particle detector database. We may then decide that our detector is inaccurate and the velocities should be subject to noise from a normal distribution, yielding a probabilistic database. This can be done measurably by the following map defined as a monad comprehension.

$$\text{addNoise} : \mathbf{B}(\text{Time} \times \text{Velocity}) \to \mathbf{GB}(\text{Time} \times \text{Velocity})$$

$$\mathsf{addNoise}(b) = \{\!|(t, v') \mid (t, v) \leftarrow \eta^{\mathrm{G}}(b), v' \leftarrow \mathbf{G}(\eta^{\mathbf{B}})(\mathsf{norm}(v, 0.1))|\!\}_{\mathbf{GB}}$$

Here we are using the respective monad units to lift $b \in \mathbf{B}X$ to $\eta^{G}(b) \in \mathbf{GB}X$ and $\mathsf{norm}(v, 0.1) \in \mathbf{G}\mathbb{R}$ to $\mathbf{G}(\eta^{\mathbf{B}})(\mathsf{norm}(v, 0.1)) \in \mathbf{GB}\mathbb{R}$.

In general, given a database in $\mathbf{B}(X \times Y)$ and some noisy transformation $f : Y \to \mathbf{G}Z$, we can generate a probabilistic database in $\mathbf{GB}(X \times Z)$ making use of the strength of the Giry monad and our distributive law $l_W : \mathbf{BG}W \to \mathbf{GB}W$.

$$\mathbf{B}(X \times Y) \xrightarrow{\mathbf{B}(X \times f)} \mathbf{B}(X \times \mathbf{G}Z) \xrightarrow{\mathbf{B}\mathrm{str}_{X,Z}} \mathbf{BG}(X \times Z) \xrightarrow{l_{X \times Z}} \mathbf{GB}(X \times Z)$$

Another use-case for random attributes, studied by Grohe and Lindner [2020], is to deal with null attributes by drawing them randomly from a vague prior distribution. This would also be easy to express using the **GB** monad.

### 6.2.2 Adding random records

We can also add and remove random records straightforwardly. Recall that using our point process monad we are easily able to define the operation of thinning on point processes (Section 4.4.2). The disjoint union operation $\uplus : \mathbf{B}X \times \mathbf{B}X \to \mathbf{B}X$ lifts to the superposition operation on point processes $\uplus : \mathbf{GB}X \times \mathbf{GB}X \to \mathbf{GB}X$ by composing with the strength of **G**. In this way the composite monad **GB** has a commutative monoid structure. Furthermore we can define point processes with some specified distribution on the number of points by regarding probability distributions on the naturals as point processes on the single point space (Section 4.4.3). By combining these ideas we can define the operation of adding and removing random points from a probabilistic database.

Suppose we have some probabilistic database in $p \in \mathbf{GB}X$, a probabilistic thinning rule $t : X \to \mathbf{GB}X$ such that it returns either a singleton or the empty bag, a distribution $d \in \mathbf{G}X$ on $X$, and some probability distribution on the naturals as a point process $\underline{N} \in \mathbf{GB}\mathbb{1}$. Then we can define an appropriate $\mathsf{addRemove} : \mathbf{GB}X \to \mathbf{GB}X$ function as follows.

$$\begin{aligned}\mathsf{addRemove}(p) = \quad & \{\!|x' \mid x \leftarrow p, x' \leftarrow t(x)|\!\}_{\mathbf{GB}} \\ & \uplus \{\!|x \mid \star \leftarrow \underline{N}, x \leftarrow \mathbf{G}\eta^{\mathbf{B}}_X(d)|\!\}_{\mathbf{GB}}\end{aligned}$$

The first line deletes random rows with some probability (the thinning rule can be

made insensitive to the input), and the second line adds in some additional random rows. Of course, a more sophisticated model could take into account other prior domain-specific information in order to decide how rows should be noisily added and removed.

## 6.3 Related work

Grohe and Lindner [2019] considered BALG[1], a subset of BALG restricted to bags of nesting level 1. That is, the queries of BALG[1] are defined on bags of type $\mathbf{B}X$ where $X$ cannot have another type $\mathbf{B}Y$ in its definition. The minimal set of queries for BALG[1] is the same set we consider here minus FLATTEN and POWERBAG since they operate on bags of bags. Grohe and Lindner showed that BALG[1] queries extend to measurable functions on bags. We have generalised their results and shown, using our monadic and monoidal structure on bags, that all of BALG extends to measurable functions on bags. This gives a clearer picture of how it comes together.

The idea of applying concepts from probabilistic programming to databases also has some momentum on the practical side, through languages such as BayesDB [Saad and Mansinghka, 2016] and PClean [Lew et al., 2021]. Slightly further afield are probabilistic logic languages such as Blog [Wu et al., 2018] and ProbLog [De Raedt and Kimmig, 2015].

[1]This is not a footnote. The subset of the query language is called BALG[1], with superscript 1.

# Chapter 7

# Conclusion

In this work we have given a categorical account of point processes by viewing them as elements of $\mathbf{G}(\mathbf{B}(X))$, i.e. as random bags. We showed that the bag endofunctor $\mathbf{B}$ is a monoidal monad, and defined a distributive law $\mathbf{BG} \to \mathbf{GB}$ using the monoidal structure, proving $\mathbf{GB}$ is a strong monad. We also considered infinite point processes as random streams $\mathbf{GS}$. Having shown $\mathbf{S}$ is a strong monad we prove that there is no distributive law $\mathbf{SG} \to \mathbf{GS}$, but that there is a natural weak law. We showed that there is a natural morphism from point processes to their intensity measures and prove that this is a morphism of monads. Finally we looked at probabilistic databases, which are examples of point processes. We presented a semantics for measurable queries on these databases making use of the bag monad and showed how the point process monad can be used to construct such databases.

## 7.1 Future work

In Chapter 4 we saw that there is no distributive law between the stream and Giry monads, but that there is a very natural weak distributive law. We conjecture that such a distributive law arises from Ionescu-Tulcea's theorem being the construction of some categorical limit in the Kleisli category of the Giry monad. It remains to be seen if this limit construction can be generalised to the composition of more general monads satisfying certain properties.

With the structure of point processes and point process operations clearly laid out, we can extend LazyPPL [Dash et al., 2023] with functions for generative

infinite point processes (random streams) and look into more optimised inference algorithms for point processes in general. There are already many useful inference algorithms present in LazyPPL, and it would be interesting to see if they can easily be lifted and extended to point processes in a natural way, such that they can make use of the point process structure to minimise wasted computations. Finally, there is a probabilistic variant of Datalog, known as GDatalog, which has recently been proposed as a generative language for probabilistic databases [Bárány et al., 2017, Grohe et al., 2020]. It may be interesting and instructive to implement GDatalog in LazyPPL using the point process monad. The semantics of non-recursive GDatalog programs can be easily given using the bag monad, just like with Balg. Then the question of the semantics of the recursive fragment may be best explored in LazyPPL.

## 7.2 Takeaways

We list some useful takeaways from this work, depending on the interests of the reader.

**For category theorists and monad enthusiasts.** There has recently been an uptick in research on the composition of monads via distributive laws, both about their existence and non-existence. In our work we have presented two such compositions, where in the case of the Giry and bag monads we obtained a distributive law, and in the case of the Giry and stream monads a weak distributive law with a proof of non-existence of a full distributive law. All of these monads were in the category of measurable spaces, which is not a cartesian closed category, indicating that interesting structures can still be recovered from such categories. Moreover, most of the work on no-go theorems for distributive laws [Zwart and Marsden, 2019] is in the setting of **Set**. Our proof of the non-existence of a distributive law between the Giry and stream monads is in **Meas**, but that is mostly because there is no stream monad on finite distributions. Our proof did not make use of any arguments specifically relating to measurable spaces, and so it may be that there is a generalisation to a no-go theorem for monads in a more general category.

**For functional programmers.** A common engineering issue that arises when programming with effects is the question of interactions between multiple effects and how to work with them in a principled and organised manner. We have

shown examples of point process constructions which work out nicely due to the combination of our distributive law along with monoidal operations on bags. There is potential for finding similar patterns in other settings of combined effects.

**For implementors of probabilistic programming languages.** Probabilistic programming lies in the intersection of programming language theory and statistical modelling. A key feature of probabilistic programming languages is that of performing inference, which is the process of computing posterior distributions from prior ones. This is a computationally intensive process, and the study of these inference algorithms is a topic of active research. From our work it is clear that there are interesting algebraic structures present in point processes. Such mathematical structures can be exploited to design inference algorithms optimised for point processes along with corresponding proofs of correctness.

# Bibliography


Aumann, R. J. (1961). Borel structures for function spaces. *Illinois Journal of Mathematics*, 5:614–630.

Beck, J. (1969). Distributive laws. In Eckmann, B., editor, *Seminar on Triples and Categorical Homology Theory*, pages 119–140, Berlin, Heidelberg. Springer Berlin Heidelberg.

Bárány, V., ten Cate, B., Kimelfeld, B., Olteanu, D., and Vagena, Z. (2017). Declarative probabilistic programming with Datalog. *ACM Transactions on Database Systems (TODS)*, 42(4).

Bochner, S. (1955). *Harmonic Analysis and the Theory of Probability*. University of California Press, Berkeley.

Ceylan, I. I., Darwiche, A., and Van den Broeck, G. (2016). Open-world probabilistic databases. In *Proc. KR 2016*.

Choksi, J. R. (1958). Inverse limits of measure spaces. *Proceedings of the London Mathematical Society*, s3-8(3):321–342.

Dahlqvist, F., Danos, V., and Garnier, I. (2016). Giry and the Machine. In *Proc. MFPS 2016*, pages 85–110.

Dahlqvist, F., Parlant, L., and Silva, A. (2018). Layer by layer – combining monads. In *Proc. ICTAC 2018*.

Daley, D. and Vere-Jones, D. (2006). *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Probability and Its Applications. Springer New York.

Dash, S., Kaddar, Y., Paquet, H., and Staton, S. (2023). Affine monads and lazy structures for bayesian programming. *Proc. ACM Program. Lang.*, 7(POPL).

Dash, S. and Staton, S. (2020). A monad for probabilistic point processes. In *Proc. ACT 2020*.

Dash, S. and Staton, S. (2021). Monads for measurable queries in probabilistic databases. In *Mathematical Foundations of Programming Semantics*.

De Raedt, L. and Kimmig, A. (2015). Probabilistic (logic) programming concepts. *Machine Learning*, 100(1):5–47.

Fremlin, D. (2000). *Measure Theory*. Number v. 1 in Measure theory. Torres Fremlin.

Gao, S. (2008). *Invariant Descriptive Set Theory*. Pure and Applied Mathematics. Chapman and Hall/CRC, 1 edition.

Giry, M. (1982). A categorical approach to probability theory. In *Categorical aspects of topology and analysis (Ottawa, Ont., 1980)*, volume 915 of *Lecture Notes in Mathematics*, pages 68–85. Springer, Berlin.

Goy, A. and Petrisan, D. (2020). Combining probabilistic and non-deterministic choice via weak distributive laws. In *Proc. LICS 2020*.

Grohe, M., Kaminski, B. L., Katoen, J.-P., and Lindner, P. (2020). Generative datalog with continuous distributions. In *Proc. PODS 2020*, pages 347–360.

Grohe, M. and Lindner, P. (2019). Probabilistic databases with an infinite open-world assumption. In *Proc. PODS 2019*, pages 17–31.

Grohe, M. and Lindner, P. (2020). Infinite probabilistic databases. In *Proc. ICDT 2020*.

Grumbach, S., Libkin, L., Milo, T., and Wong, L. (1996). Query languages for bags: expressive power and complexity. *SIGACT News (Database Theory Column)*, pages 30–37.

Halmos, P. (1976). *Measure Theory*. Graduate Texts in Mathematics. Springer New York.

Herlau, T., Schmidt, M. N., and Morup, M. (2016). Completely random measures for modelling block-structured sparse networks. In *Proc. NeurIPS 2016*, pages 4260–4268.

Heunen, C., Kammar, O., Staton, S., and Yang, H. (2017). A convenient category for higher-order probability theory. In *Proc. LICS 2017*. IEEE Press.

Hutton, G. (1999). A tutorial on the universality and expressiveness of fold. *J. Funct. Program.*, 9(4):355–372.

Hyland, M., Nagayama, M., Power, J., and Rosolini, G. (2006). A category theoretic formulation for engeler-style models of the untyped $\lambda$-calculus. *Electronic notes in theoretical computer science*, 161:43–57.

Ionescu-Tulcea, C. (1949). Mesures dans les espaces produits. *Atti Accad. Naz. Lincei Rend.*, 7:208–211.

Jacobs, B. (2019a). Learning along a channel:the expectation part of expectation-maximisation. In *Proc. MFPS 2019*.

Jacobs, B. (2019b). Structured probabilitistic reasoning. Draft available from the author's website.

Jacobs, B. and Staton, S. (2020). De Finetti's construction as a categorical limit. In *Proc. CMCS 2020*.

Jampani, R., Xu, F., Wu, M., Perez, L. L., Jermaine, C., and Haas, P. J. (2008). MCDB: A Monte Carlo approach to managing uncertain data. In *Sigmod 2008*.

Kechris, A. (1995). *Classical Descriptive Set Theory*. Graduate Texts in Mathematics. Springer New York.

Keimel, K. and Plotkin, G. (2017). Mixed powerdomains for probability and nondeterminism. arXiv:1612.01005.

Klug, A. (1982). Equivalence of relational algebra and relational calculus query languages having aggregate functions. *J. ACM*, 29(3):699–717.

Kock, A. (1970). Monads on symmetric monoidal closed categories. *Archiv der Mathematik*, 21:1–10.

Last, G. and Penrose, M. (2017). *Lectures on the Poisson Process*. Institute of Mathematical Statistics Textbooks. Cambridge University Press.

Lew, A. K., Agrawal, M., Sontag, D., and Mansinghka, V. K. (2021). PClean: Bayesian data cleaning at scale with domain-specific probabilistic programming. In *Proc. AISTATS 2021*.

Macchi, O. (1975). The coincidence approach to stochastic point processes. *Advances in Applied Probability*, 7:83–122.

Manes, E. and Mulry, P. (2007). Monad compositions I: general constructions and recursive distributive laws. *Theory and Applications of Categories*, 18:172–208.

Matérn, B. (1960). *Spatial variation: Stochastic models and their application to some problems in forest surveys and other sampling investigations.*

Moggi, E. (1991). Notions of computation and monads. *Inf. Comput.*, 93(1):55–92.

Moyal, J. E. (1962). The general theory of stochastic population processes. *Acta Mathematica*, 108.

Özsoyoğlu, G., Özsoyoğlu, Z. M., and Matos, V. (1987). Extending relational algebra and relational calculus with set-valued attributes and aggregate functions. *ACM Trans. Database Syst.*, 12(4):566–592.

Peyton Jones, S. and Wadler, P. (2007). Comprehensive comprehensions. In *Proc. Haskell 2007*.

Rao, M. (1971). Projective limits of probability spaces. *Journal of Multivariate Analysis*, 1(1):28–57.

Saad, F. and Mansinghka, V. (2016). A probabilistic programming approach to probabilistic data analysis. In *NeurIPS*.

Thomas, M. (1949). A generalization of poisson's binomial limit for use in ecology. *Biometrika*, 36:18–25.

Varacca, D. and Winskel, G. (2006). Distributing probability over non-determinism. *Mathematical structures in computer science*, 16:87–113.

Wadler, P. (1992). Comprehending monads. *Mathematical Structures in Computer Science*, 2:461–493.

Wald, A. (1944). On Cumulative Sums of Random Variables. *The Annals of Mathematical Statistics*, 15(3):283 – 296.

Wilf, H. S. (2006). *Generatingfunctionology*. A. K. Peters, Ltd., USA.

Wu, Y., Srivastava, S., Hay, N., Du, S., and Russell, S. J. (2018). Discrete-continuous mixtures in probabilistic programming: Generalized semantics and inference algorithms. In *Proc. ICML 2018*, pages 5339–5348.

Zwart, M. and Marsden, D. (2019). No-go theorems for distributive laws. In *Proc. LICS 2019*.

# Appendix A

# Monadic point process constructions in LazyPPL

Below are all the point process constructions defined in Chapter 4, written in Haskell. We define the Giry monad as `Prob`, taken from the probabilistic programming library `LazyPPL`. To keep things simple we define bags to be lists, bearing in mind that elements in lists are unordered. Finally we also assume that all functions being folded over are symmetric functions.

```
{-# LANGUAGE MonadComprehensions #-}

import LazyPPL

import Control.Applicative
import Control.Monad (liftM, ap)
import Data.List hiding (filter)
import Prelude hiding (pi, filter)
import Statistics.Distribution
import Statistics.Distribution.Normal (normalDistr)
import System.Random hiding (uniform)
import qualified Statistics.Distribution.Poisson as Poisson

import Text.Printf (printf)

data Bag a = Bag [a] deriving (Functor, Show)
```

```
unBag (Bag xs) = xs

data GB a = GB (Prob (Bag a)) deriving (Functor)

unGB (GB p) = p

instance Applicative GB where
  pure  = \x -> GB $ Prob $ const (Bag [x])
  (<*>) = ap

instance Applicative Bag where
  pure  = \x -> Bag [x]
  (<*>) = ap

fold :: ((x,y) -> y) -> ((Bag x, y) -> y)
fold f (Bag [], y) = y
fold f (Bag (x:xs), y) = f (x, fold f (Bag xs, y))

dst :: (Prob x, Prob y) -> Prob (x,y)
dst (p1,p2) = [ (x,y) | x <- p1, y <- p2 ]

add :: (x, Bag x) -> Bag x
add (x, Bag xs) = Bag (x : xs)

distrAcc :: (Prob x, GB x) -> GB x
distrAcc (p, GB a) = GB $ fmap add $ dst (p,a)

distLaw :: Bag (Prob a) -> GB a
distLaw bg = fold distrAcc (bg, GB $ return $ Bag [])

joinB :: Bag (Bag a) -> Bag a
joinB (Bag []) = Bag []
joinB (Bag (x:xs)) = Bag $ (unBag x) ++ (unBag $ joinB $ Bag xs)

joinGB :: GB (GB a) -> GB a
joinGB (GB q) = GB $ fmap joinB $ muG $ fmap unGB $ fmap distLaw p
  where
    p = fmap (fmap unGB) q
    muG :: Prob (Prob a) -> Prob a
    muG p = p >>= id

instance Alternative Bag where
  empty = Bag []
```

```
  (<|>) = curry bunion

instance Monad Bag where
  return = pure
  p >>= f = joinB $ fmap f p

instance Monad GB where
  return = pure
  p >>= f = joinGB $ fmap f p

sampleGB :: GB x -> IO (Bag x)
sampleGB (GB p) = do newStdGen
                     g <- getStdGen
                     let rs = randomTree g
                     return $ runProb p rs

-- | 4.4.1 Probability distributions as point processes

-- | Probability distributions on the naturals

natToBag :: Int -> Bag ()
natToBag n = Bag $ replicate n ()

poisson :: Double -> Prob Int
poisson lambda = do
  x <- uniform
  let cmf = scanl1 (+) $
            map (probability $ Poisson.poisson lambda) [0,1..]
  let (Just n) = findIndex (> x) cmf
  return $ n

poisDistr :: Double -> GB ()
poisDistr l = GB $ fmap natToBag (poisson l)

-- > sampleGB $ poisDistr 5
-- Bag [(),(),(),()]

-- > sampleGB $ poisDistr 5
-- Bag [(),(),(),(),(),(),()]

-- | Building compound probability distributions

-- The iid samples are X ~ Pois 2 and we add N ~ Pois 5 of them.
```

```
compound :: GB ()
compound = [ () | () <- poisDistr 5, () <- poisDistr 2 ]

-- | 4.4.2 Point process operations

-- | Superposition

bunion :: (Bag x, Bag x) -> Bag x
bunion (b1, b2) = fold add (b1, b2)

superposition :: (GB x, GB x) -> GB x
superposition (GB p1, GB p2) = GB $ [ bunion (b1,b2) | b1 <- p1,
                                                       b2 <- p2 ]

-- | Thinning a point process

thinP :: (x -> GB x) -> GB x -> GB x
thinP thinRule gamma = [ x' | x <- gamma, x' <- thinRule x ]

-- | Displacing a point process

displace :: Prob Double -> GB Double -> GB Double
displace delta gamma = [ x + d | x <- gamma,
                                 d <- GB $ fmap return delta ]

-- | 4.4.3 Constructing point processes

-- | Poisson point processes

unif :: Prob Double
unif = uniform

u2 :: GB (Double, Double)
u2 = GB [ return (x,y) | x <- unif, y <- unif ]

-- > sampleGB $ u2
-- Bag [(0.9061518166629333,0.4672754088297997)]

-- > sampleGB $ u2
-- Bag [(0.5232338666399926,0.6247643232136042)]

pi :: Double -> GB (Double, Double)
```

```
pi l = [ (x,y) | () <- poisDistr l, (x,y) <- u2 ]

-- > sampleGB $ pi 10
-- Bag [(0.7755897094265392,0.260628180304715),
--      (0.6753237064797476,0.9635003310390328),
--      ...
--      (0.3342172615555511,0.2607043902602101),
--      (0.6552245376447139,7.877889689316608e-2)]

-- | Clustered point processes

normal :: Double -> Double -> Prob Double
normal m s = do
  x <- uniform
  return $ quantile (normalDistr m s) x

tau :: Double -> Double -> GB (Double, Double)
tau l1 l2 = [ (x,y) | (x',y') <- pi l1, () <- poisDistr l2,
                      x <- GB $ fmap return (normal x' 0.1),
                      y <- GB $ fmap return (normal y' 0.1) ]

-- > fmap $ sampleGB $ tau 6 3
-- Bag [(0.5301574945514493,0.6341422035000739),
--      (0.42366480608025847,0.8472570699434515),
--      ...
--      (0.28200849338624046,0.673606124431263),
--      (0.2879842971912776,0.7958024520418397)]

-- | Matern hard-core point processes

-- All of the functions below are measurable with respect to
-- standard Borel spaces.

filter :: (x -> Bool) -> Bag x -> Bag x
filter p b = [ x | x <- b, p x ]

allB :: (x -> Bool) -> Bag x -> Bool
allB p b = fold (\(x,y) -> p x && y) (b, True)

t :: Double -> ((Double,Double), Bag (Double,Double))
     -> Bag (Double,Double)
t r (x,b) = if allB (== True) $ fmap (\y -> dist x y > r) b
            then Bag [x]
```

```
            else Bag []
  where
    dist :: (Double,Double) -> (Double,Double) -> Double
    dist (a,b) (c,d) = sqrt ((a-c)^2 + (b-d)^2)

thin :: Double -> Bag (Double,Double) -> Bag (Double,Double)
thin r b = [ x | y <- b, x <- t r (y, filter (/= y) b) ]

matern :: Double -> Double -> GB (Double,Double)
matern r l = GB $ fmap (thin r) (unGB $ pi 10)

-- > sampleGB $ matern 0.1 15
-- Bag [(0.32994922704072593,0.3986441098534481),
--      ...
--      (0.7592779520866937,0.48555201403031867),
--      (0.4205573008130199,0.5899135568737532)]
```