Infinite Probabilistic Databases

Martin Grohe
RWTH Aachen University, Germany
grohe@informatik.rwth-aachen.de

Peter Lindner
RWTH Aachen University, Germany
lindner@informatik.rwth-aachen.de

Abstract

Probabilistic databases (PDBs) are used to model uncertainty in data in a quantitative way. In the standard formal framework, PDBs are finite probability spaces over relational database instances. It has been argued convincingly that this is not compatible with an open-world semantics (Ceylan et al., KR 2016) and with application scenarios that are modeled by continuous probability distributions (Dalvi et al., CACM 2009).

We recently introduced a model of PDBs as infinite probability spaces that addresses these issues (Grohe and Lindner, PODS 2019). While that work was mainly concerned with countably infinite probability spaces, our focus here is on uncountable spaces. Such an extension is necessary to model typical continuous probability distributions that appear in many applications. However, an extension beyond countable probability spaces raises nontrivial foundational issues concerned with the measurability of events and queries and ultimately with the question whether queries have a well-defined semantics.

It turns out that so-called finite point processes are the appropriate model from probability theory for dealing with probabilistic databases. This model allows us to construct suitable (uncountable) probability spaces of database instances in a systematic way. Our main technical results are measurability statements for relational algebra queries as well as aggregate queries and Datalog queries.

2012 ACM Subject Classification Mathematics of computing → Probabilistic representations; Theory of computation → Incomplete, inconsistent, and uncertain databases; Theory of computation → Database query languages (principles)

Keywords and phrases Probabilistic Databases, Possible Worlds Semantics, Query Measurability, Relational Algebra, Aggregate Queries

Digital Object Identifier 10.4230/LIPIcs.ICDT.2020.16

Related Version Full version available on arXiv: `https://arxiv.org/abs/1904.06766`.

Funding *Peter Lindner*: This author is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Research Training Group 2236 UnRAVeL.

Acknowledgements We are grateful to Sam Staton for insightful discussions related to this work, and for pointing us to point processes. We also thank Peter J. Haas for discussions on the open-world assumption and the math behind the MCDB system.

1 Introduction

Probabilistic databases (PDBs) are used to model uncertainty in data. Such uncertainty could be introduced by a variety of reasons like, for example, noisy sensor data, the presence of incomplete or inconsistent information, or because the information is gathered from unreliable sources [3, 63]. In the standard formal framework, probabilistic databases are finite probability spaces whose sample spaces consist of database instances in the usual sense, referred to as “possible worlds”. However, this framework has various shortcomings due

23rd International Conference on Database Theory (ICDT 2020).
Editors: Carsten Lutz and Jean Christoph Jung; Article No. 16; pp. 16:1–16:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to its inherent *closed-world assumption* [16] – in particular, any event outside of the finite scope of such probabilistic databases is treated as an impossible event. There is also work on PDBs that includes continuous probability distributions and hence goes beyond the formal framework of finite probability space. Yet, these continuous PDBs lack a general formal basis in terms of a possible worlds semantics [20]. While both open-world PDBs and continuous probability distributions in PDBs have received some attention in the literature, there is no systematic joint treatment of these issues with a sound theoretical foundation. In [38], we introduced an extended model of PDBs as arbitrary (possibly infinite) probability spaces over finite database instances. However, the focus there was on countably infinite PDBs. An extension to continuous PDBs, which is necessary to model probability distributions appearing in many applications that involve real-valued measurement data, raises new fundamental questions concerning the measurability of events and queries.

In this paper, we lay the foundations of a systematic and sound treatment of infinite, even uncountable, probabilistic databases, and we prove that queries expressed in standard query languages have a well-defined semantics.

Our treatment is based on the mathematical theory of finite point processes [53, 49, 18]. Adopting this theory to the context of relational databases, we give a suitable construction of measurable spaces over which our probabilistic databases can then be defined. The only assumption that we need to make is that the domains of all attributes satisfy certain topological assumptions (they need to be Polish spaces; all standard domains such as integers, strings, reals, satisfy this assumption). For queries and views to have a well-defined open-world semantics, we need them to be measurable mappings between probabilistic databases. Our main technical result states that indeed all queries and views that can be expressed in the relational algebra, even equipped with arbitrary aggregate operators (satisfying some mild measurability conditions) are measurable mappings. The result holds for both a bag-based and set-based relational algebra. We also prove the measurability of Datalog queries.

Measurability of queries may seem like an obvious minimum requirement, but one needs to be very careful. We give an example of a simple, innocent looking "query" that is not measurable (see Example 8). The proofs of the measurability results are not trivial, which may already be seen from the fact that they depend on the topological assumption that the attribute domains are Polish spaces (most importantly, they are complete topological spaces and have a countable dense subset). At their core, the proofs are based on finding suitable "countable approximations" of the queries.

In the last section of this paper, we briefly discuss queries for probabilistic databases that go beyond "standard" database queries lifted to probabilistic databases via an open-world semantics. Examples of such a queries are probabilistic threshold queries and rank queries. Such queries refer not only to the facts in a database, but also to their probabilities, and hence are inherently probabilistic.

Related Work. Early work on models for probabilistic databases dates back to the 1980s [69, 35, 15] and 1990s [8, 56, 26, 34, 71]. These models may be seen as special cases or variations of the now-acclaimed formal model of probabilistic databases that features a usually finite set of database instances (the "possible worlds") together with a probability distribution among them [3, 63].

The work [45] presents a formal definition of the probabilistic semantics of relational algebra queries as it is used in the MayBMS system [46]. A probabilistic semantics for Datalog has already been proposed in the mid-90s [33]. More recently, a version of Datalog was considered in which rules may fire probabilistically [25]. Aggregate queries in probabilistic databases were first treated systematically in [59] and reappear in various works concerning particular PDB systems [54, 28].

The models of possible worlds semantics mentioned above are the mathematical backbone of existing probabilistic database prototype systems such as MayBMS [46], Trio [68] and MystiQ [12]. Various subsequent prototypes feature uncountable domains as well, such as Orion [60], MCDB [41, 42], new versions of Trio [4] and PIP [44]. The MCDB system in particular allows programmers to specify probabilistic databases with infinitely many possible worlds with database instances that can grow arbitrarily large [42] and is therefore probably the most general existing system. Its system-driven description does not feature a general formal, measure theoretic account of its semantics though. In a spirit that is similar to our presentation here, the work [64] introduced a measure theoretic semantics for probabilistic data stream systems with probability measures composed from Gaussian mixture models but (to our knowledge) on a per tuple basis and without the possibility of inter-tuple correlations. Continuous probabilistic databases have already been considered earlier in the context of sensor networks [27, 17, 24]. The first work to formally introduce continuous possible worlds semantics (including aggregation) is [1] for probabilistic XML. However, the framework has an implicit restriction bounding the number of tuples in a PDB.

Models similar in expressivity to the one we present have also been suggested in the context of probabilistic modeling languages and probabilistic programming [51, 52, 58, 22, 7]. In particular notable are the measure theoretic treatments of Bayesian Logic (BLOG) [51] in [70] and Markov Logic Networks (MLNs) [58] in [61]. While these data models are relational, it is unclear, how suitable they are for general database applications and in particular, the investigation of typical database queries is beyond the scope of these works.

Problems raised by the closed-world assumption [57] in probabilistic databases was discussed initially by Ceylan et al. in [16] where they suggest the model of OpenPDBs. In [10], the authors make a more fine-grained distinction between an *open-world* and *open-domain assumption*, the latter of which does not assume the attribute values of the database schema to come from a known finite domain. The work [31] considers semantic constraints on open worlds in the OpenPDB framework. The semantics of OpenPDBs can be strengthened towards an open-domain assumption by the means of ontologies [9, 10, 11].

The classification of views we discuss towards the end of this paper shares similarities with previous classifications of queries such as [17] in the sense that it distinguishes *how* aggregation is involved. The work [66] suggests a distinction between "traditional" and "out-of-world aggregation" quite similar to the one we present.

2 Preliminaries

Throughout the paper, we denote the set of nonnegative integers by $\mathbb{N}$, the set of rational numbers by $\mathbb{Q}$ and the set of real numbers by $\mathbb{R}$. We write $\mathbb{N}_+$, $\mathbb{Q}_+$ and $\mathbb{R}_+$ for the restrictions of these sets to strictly positive numbers.

If M is a set and $k \in \mathbb{N}$, then $\binom{M}{k}$ denotes the set of subsets of M of cardinality k. The set of all finite subsets of M is then given by $\bigcup_{k\geq 0}\binom{M}{k} =: \binom{M}{<\omega}$.

A *bag* (also called *multiset*) over a set U is an unordered collection of elements of U, possibly with repetitions. In order to distinguish sets and bags, we use double curly braces $\{\!\{\cdots\}\!\}$ when explicitly denoting bags. Similarly to the notation for sets, we let $(\!\binom{M}{k}\!)$ denote the set of bags over the set M of cardinality $k \in \mathbb{N}$ (that is, containing k elements, counting copies). The set of all finite bags over M is given by $\bigcup_{k\geq 0}(\!\binom{M}{k}\!) =: (\!\binom{M}{<\omega}\!)$.

There are multiple equivalent ways to formalize the notion of bags. We introduce two such definitions that we use interchangeably later:

Multiplicity perspective A *bag* B over some set U is a function $\#_B \colon U \to \mathbb{N}$ assigning a *multiplicity* to every element of U. The cardinality of B is $|B| \coloneqq \sum_{u \in U} \#_B(u)$.

Quotient perspective For all $a, b \in U^k$, let $a \sim b$ if b is a permutation of a. A *bag* B of cardinality $|B| = k$ is a $\sim$-equivalence class on U^k.

While the multiplicity perspective better matches the intuitive semantics of bags, the quotient view later has a closer connection to the probability spaces we are going to construct.

2.1 Relational Databases

We follow the general terminology and notions of the *named perspective* of databases, see for example [2]. We fix two countably infinite, disjoint sets **Attributes** and **Relations** of *attribute names* and *relation names*, respectively. As usual, we drop the distinction between names of attributes and relations and their model-theoretic interpretation. A *database schema* is a pair $\mathcal{S} = (\mathcal{A}, \mathcal{R})$ with the following properties:

- $\mathcal{A}$ and $\mathcal{R}$ are finite subsets of **Attributes** resp. **Relations**.
- For every attribute $A \in \mathcal{A}$ there exists a set $\mathsf{dom}_{\mathcal{S}}(A)$, called its *domain*.
- For every relation symbol $R \in \mathcal{R}$ there exists an associated k-tuple of distinct attributes from $\mathcal{A}$ for some k, called its *type* $\mathsf{type}_{\mathcal{S}}(R)$.

Implicitly, every relation $R \in \mathcal{R}$ has an *arity* $\mathsf{ar}_{\mathcal{S}}(R) \coloneqq |\mathsf{type}_{\mathcal{S}}(R)|$ and a *domain* $\mathsf{dom}_{\mathcal{S}}(R) \coloneqq \prod_{A \in \mathsf{type}_{\mathcal{S}}(R)} \mathsf{dom}_{\mathcal{S}}(A)$. Elements of the domain of $R \in \mathcal{R}$ are called *R-tuples.* Whenever a pair $(\mathcal{A}, \mathcal{R})$ is given, we assume that all of the aforementioned mappings are given as well, unless it is specified otherwise. Given a database schema $\mathcal{S} = (\mathcal{A}, \mathcal{R})$ and a relation $R \in \mathcal{R}$, the set of *R-facts* in $\mathcal{S}$ is formally defined as $\mathsf{facts}_{\mathcal{S}}(R) = \{R\} \times \mathsf{dom}_{\mathcal{S}}(R)$. The set of *all* facts of schema $\mathcal{S}$ is given as $\mathsf{facts}_{\mathcal{S}}(\mathcal{R}) \coloneqq \bigcup_{R \in \mathcal{R}} \mathsf{facts}_{\mathcal{S}}(R)$.

As usual, we denote R-facts in the fashion of $R(a_1, \dots, a_k)$ rather than $(R, a_1, \dots, a_k)$. If $U \subseteq \mathsf{dom}_{\mathcal{S}}(R)$ for $R \in \mathcal{R}$, we let $R(U) \coloneqq \{R(u) \colon u \in U\}$. If U is a Cartesian product involving singletons, like for example $U = \{a\} \times V$, we may omit the braces of the singletons and replace crosses with commas so that $R(a, U) = \{R(a, u) \colon u \in U\}$.

Finally, a *database instance* D of schema $\mathcal{S} = (\mathcal{A}, \mathcal{R})$ is a *finite* bag of facts from $\mathsf{facts}_{\mathcal{S}}(\mathcal{R})$, that is, an element of the set $\mathbb{D}_{\mathcal{S}} \coloneqq \left(\!\!\binom{\mathsf{facts}_{\mathcal{S}}(\mathcal{R})}{<\omega}\!\!\right)$. We want to emphasize that in particular we allow single facts to appear two or more times within an instance. That is, we use bag semantics in our database instances.

2.2 Topology and Measure Theory

We assume that the reader is familiar with the basic notions of point set topology such as open and closed sets and continuous mappings. For a more detailed introduction to the concepts we refer to standard text books such as [14, 13]. In the following, we concentrate on the background from measure theory. The definitions and statements are based upon [62] and Chapter 1 of [43].

In topological terms, the spaces we use as our attribute domains later on are called Polish spaces - complete, separable metrizable spaces. Such spaces are the default choice for probability theory in a general setting, as they are quite general while still exhibiting the nice behavior of closed intervals of the real line, in particular the ability to approximate points by converging sequences of a countable collection of open sets.

▶ **Example 1** (see [32, ch. 18] and [62, pp. 52 et seqq.]).

- All finite and countably infinite spaces (with the discrete topology) are Polish.
- The spaces $\mathbb{R}$ and $\mathbb{R} \cup \{\pm\infty\}$ are Polish.
- Closed subspaces of Polish spaces are Polish.
- Countable disjoint unions and countable products of Polish spaces are Polish.

These examples already capture the most relevant cases for standard database applications. Nevertheless we stick to the abstract notion of Polish spaces in order to keep the framework as general as possible. When we work with Polish spaces, we will later always assume that we work with a fixed metric on the space (turning it into a complete separable metric space). In particular, we will use the standard notation $B_\varepsilon(x)$ for the *ball of radius* ε around the point x (with respect to said metric).

Let $\mathbb{X}$ be some set. A *σ-algebra* on $\mathbb{X}$ is a family $\mathfrak{X}$ of subsets of $\mathbb{X}$ such that $\mathbb{X} \in \mathfrak{X}$ and $\mathfrak{X}$ is closed under complementation and *countable* unions. If $\mathfrak{G}$ is a family of subsets of $\mathbb{X}$, then the *σ-algebra generated by* $\mathfrak{G}$ is the smallest σ-algebra $\mathfrak{X}$ on $\mathbb{X}$ containing $\mathfrak{G}$. A *measurable space* is a pair $(\mathbb{X}, \mathfrak{X})$ where $\mathbb{X}$ is an arbitrary set and $\mathfrak{X}$ is a σ-algebra on $\mathbb{X}$. Subsets of $\mathbb{X}$ are called *$\mathfrak{X}$-measurable* (or *measurable* if $\mathfrak{X}$ is clear from context) if they belong to $\mathfrak{X}$. A *probability measure* on $\mathbb{X}$ is a countably additive function $P\colon \mathfrak{X} \to [0,1]$ with $P(\emptyset) = 0$ and $P(\mathbb{X}) = 1$. (P being countably additive means $P\big(\bigcup_i \mathcal{X}_i\big) = \sum_i P(\mathcal{X}_i)$ for any sequence $\mathcal{X}_0, \mathcal{X}_1, \mathcal{X}_2, \dots$ of disjoint measurable sets.) A measurable space equipped with a probability measure is called a probability space. If Ξ is a probability space $(\mathbb{X}, \mathfrak{X}, P)$, we also write $\mathsf{Pr}_{X \sim \Xi}(X \in \mathcal{X}) = P(\mathcal{X})$ or even omit the subscript $X \sim \Xi$, if the underlying probability space is clear from context.

Let $(\mathbb{X}, \mathfrak{X})$ and $(\mathbb{Y}, \mathfrak{Y})$ be measurable spaces. A mapping $\varphi\colon \mathbb{X} \to \mathbb{Y}$ is called $(\mathfrak{X}, \mathfrak{Y})$-*measurable* (or simply *measurable* if the involved σ-algebras are clear from context) if the preimage under φ of every $\mathfrak{Y}$-measurable set is $\mathfrak{X}$-measurable. That is, if

$$\varphi^{-1}(\mathcal{Y}') = \{X \in \mathbb{X}\colon \varphi(X) \in \mathcal{Y}'\} \in \mathfrak{X} \qquad \text{for all } \mathcal{Y}' \in \mathfrak{Y}.$$

▶ **Fact 2** (cf. [43, Lemmas 1.4, 1.7 & 1.10]). *Let $(\mathbb{X}, \mathfrak{X})$, $(\mathbb{Y}, \mathfrak{Y})$, $(\mathbb{Z}, \mathfrak{Z})$ be measurable spaces.*

- *Let $\mathfrak{G}$ generate $\mathfrak{Y}$. If $\varphi\colon \mathbb{X} \to \mathbb{Y}$ satisfies $\varphi^{-1}(\mathcal{G}) \in \mathfrak{X}$ for all $\mathcal{G} \in \mathfrak{G}$, then φ is measurable.*
- *If $\varphi\colon \mathbb{X} \to \mathbb{Y}$ and $\psi\colon \mathbb{Y} \to \mathbb{Z}$ are measurable, then $\psi \circ \varphi\colon \mathbb{X} \to \mathbb{Z}$ is $(\mathfrak{X}, \mathfrak{Z})$-measurable.*
- *If $\mathbb{Y}$ is a metric space and $(\varphi_n)_{n \geq 0}$ is a sequence of measurable functions $\varphi_n\colon \mathbb{X} \to \mathbb{Y}$ with $\lim_{n\to\infty} \varphi_n = \varphi$, then φ is measurable as well.*

If $(\mathbb{X}, \mathfrak{T}_\mathbb{X})$ is a topological space, the *Borel σ-algebra* $\mathfrak{Bor}_\mathbb{X}$ on $\mathbb{X}$ is the σ-algebra generated by $\mathfrak{T}_\mathbb{X}$. Sets in the Borel σ-algebra are also called *Borel*.

▶ **Fact 3** (cf. [43, Lemma 1.5]). *Any continuous function between the topological spaces $(\mathbb{X}, \mathfrak{T}_\mathbb{X})$ and $(\mathbb{Y}, \mathfrak{T}_\mathbb{Y})$ is $(\mathfrak{Bor}_\mathbb{X}, \mathfrak{Bor}_\mathbb{Y})$-measurable .*

Two measurable spaces $(\mathbb{X}, \mathfrak{X})$ and $(\mathbb{Y}, \mathfrak{Y})$ are called *isomorphic* if there exists a bijection $\varphi\colon \mathbb{X} \to \mathbb{Y}$ such that both φ and φ^{-1} are measurable. The mapping φ is then called an *isomorphism* between the measurable spaces. If $\mathfrak{X} = \mathfrak{Bor}_\mathbb{X}$ and $\mathfrak{Y} = \mathfrak{Bor}_\mathbb{Y}$, then φ is called a *Borel isomorphism* and the measurable spaces are called *Borel isomorphic*. Measurable spaces that are isomorphic to some Polish space with its Borel σ-algebra are called *standard Borel spaces.*

If $\mathfrak{X}_i$ is a σ-algebra on X_i for all $i \in I$, the *product σ-algebra* $\bigotimes_{i \in I} \mathfrak{X}_i$ of $(\mathfrak{X}_i)_{i \in I}$ is the σ-algebra on $\prod_{i \in I} \mathbb{X}_i$ that is generated by the sets $\{\pi_j^{-1}(\mathcal{X})\colon \mathcal{X} \in \mathfrak{X}_j\}_{j \in I}$ where π_j is the canonical projection map $\pi_j\colon \prod_{i \in I} \mathbb{X}_i \to \mathbb{X}_j$.

▶ **Fact 4** (cf. [43, Lemma 1.2]). *Let* $(\mathbb{X}_i)_{i \in I}$ *be a* countable *sequence of Polish spaces and let* $\mathfrak{Bor}_i$ *be the Borel* σ*-algebra of* $\mathbb{X}_i$. *Then* $\mathbb{X} = \prod_{i \in I} \mathbb{X}_i$ *is Polish and* $\mathfrak{Bor}_{\mathbb{X}} = \bigotimes_{i \in I} \mathfrak{Bor}_i$. *That is, countable products of standard Borel spaces are standard Borel.*

2.3 (Finite) Point Processes

Point processes are a well-known concept in probability theory that is used to model distributions of a discrete (but unknown or even infinite) number of points in some abstract "state space", say the Euclidean space $\mathbb{R}^n$ [18]. They are used to model a variety of both practical and theoretical problems and appear in a broad field of applications such as, for example, particle physics, ecology, geostatistics, astronomy and tracking [53, 18, 6, 50, 23]. A concrete collection of points that is obtained by a draw from such a distribution model is called a *realization* of the point process. If all realizations are finite, we speak of a finite point process [18]. We proceed to construct a finite point process over a Polish state space, following the classic constructions of [53, 49]. While modern point process theory is much more evolved by casting point processes in the more general framework of random measures [19], the seminal model of [53, 49] suffices for our studies due to our restriction to finite point processes.

Let $(\mathbb{X}, \mathfrak{X})$ be a standard Borel space. Then for every n, the product measurable space $(\mathbb{X}^n, \mathfrak{X}^{\otimes n})$ with $\mathfrak{X}^{\otimes n} := \mathfrak{X} \otimes \cdots \otimes \mathfrak{X}$ (n times) is standard Borel as well (Fact 4). Letting $\sim_n$ denote the equivalence relation on $\mathbb{X}^n$ with $(x_1, \dots, x_n) \sim_n (y_1, \dots, y_n)$ if there exists a permutation π of $\{1, \dots, n\}$ with $(y_1, \dots, y_n) = (x_{\pi(1)}, \dots, x_{\pi(n)})$, then elements of $\mathbb{X}^n / \sim_n$ are basically unordered collections of n (not necessarily different) points, that is, *bags* (or *multisets*). Formally, we identify $\mathbb{X}^n / \sim_n$ with the space $\left(\!\!\binom{\mathbb{X}}{n}\!\!\right)$ of all n-element bags from X. The space of all possible realizations is then naturally defined as

$$\left(\!\!\binom{\mathbb{X}}{<\omega}\!\!\right) = \bigcup_{n \in \mathbb{N}} \left(\!\!\binom{\mathbb{X}}{n}\!\!\right) = \bigcup_{n \in \mathbb{N}} \mathbb{X}^n / \sim_n .$$

This is the canonical sample space for a finite point process [18, 53], but we need to define a σ-algebra on this space. The original construction of [53] considers the *symmetrization* transformation sym from $\mathbb{X}^{<\omega}$ to $\left(\!\!\binom{\mathbb{X}}{<\omega}\!\!\right)$ where $\mathsf{sym}(x_1, \dots, x_n) = [(x_1, \dots, x_n)]_{\sim_n} = \{\!\!\{ x_1, \dots, x_n \}\!\!\}$ and $\mathsf{sym}(\mathcal{X}) = \{\mathsf{sym}(\bar{x}) \colon \bar{x} \in \mathcal{X}\}$ and defines the σ-algebra on $\mathbb{X}$ to be the set of all subsets of $\left(\!\!\binom{\mathbb{X}}{<\omega}\!\!\right)$ whose preimage under sym is measurable with respect to the σ-algebra on $\mathbb{X}^{<\omega}$ that is generated using $(\mathfrak{X}^{\otimes n})_{n \in \mathbb{N}}$ (pursuing the idea to lift probability measures from well-known product spaces to the new, in terms of measure theory inconvenient "bag-space" – note that the construction above indeed yields a σ-algebra on $\left(\!\!\binom{\mathbb{X}}{<\omega}\!\!\right)$, see [43, Lemma 1.3]). An equivalent, but technically more convenient construction (see [49]) is motivated by an interpretation of point processes as "random counting measures" [53, 49, 19]: for $\mathcal{X} \in \mathfrak{X}$ and $n \in \mathbb{N}$, the set $\mathcal{C}(\mathcal{X}, n) \subseteq \left(\!\!\binom{\mathbb{X}}{<\omega}\!\!\right)$ is the set of bags C over $\mathbb{X}$ with $\#_C(\mathcal{X}) := \sum_{X \in \mathcal{X}} \#_C(X) = n$ (that is, with exactly n "hits" in $\mathcal{X}$) is called the *counting event* of $\mathcal{X}$ and n. We define $\mathfrak{C}_{\mathbb{X}}$ to be the σ-algebra that is generated by the family of counting events $\mathcal{C}(\mathcal{X}, n)$ where $\mathcal{X}$ is Borel in $\mathbb{X}$ and n is a nonnegative integer. The family $\mathfrak{C}_{\mathbb{X}}$ is known as the *counting σ-algebra* on $\left(\!\!\binom{\mathbb{X}}{<\omega}\!\!\right)$. It can be shown that the σ-algebra generated by the counting events is the same as the σ-algebra defined from product σ-algebras and the symmetrization operation (see [53, 49]).

▶ **Definition 5** (cf. [49, Def. 1]). *Let* $(\mathbb{X}, \mathfrak{X})$ *be a standard Borel space and let* P *be a probability measure on* $\left(\left(\!\!\binom{\mathbb{X}}{<\omega}\!\!\right), \mathfrak{C}_{\mathbb{X}}\right)$. *Then* $\left(\left(\!\!\binom{\mathbb{X}}{<\omega}\!\!\right), \mathfrak{C}_{\mathbb{X}}, P\right)$ *is called a* finite point process *with* state space $(\mathbb{X}, \mathfrak{X})$.

A finite point process $(\mathbb{Y}, \mathfrak{Y}, P)$ *with state space* $(\mathbb{X}, \mathfrak{X})$ *is called* simple, *if any realization is almost surely a set, i. e. if* $\mathsf{Pr}\left(\#_Y\left(\{X\}\right) \in \{0, 1\} \text{ for all } X \in \mathbb{X}\right) = 1$.

3 Probabilistic Databases

In [38], we introduced a general notion of infinite probabilistic databases as probability spaces of database instances, that is, probability spaces $(\mathbb{D}, \mathfrak{D}, P)$, where $\mathbb{D} \subseteq \mathbb{D}_{\mathcal{S}}$ for some database schema $\mathcal{S}$. Here $\mathbb{D}$ may be infinite, even uncountable. In fact, in [38] we only considered instances that are sets rather than bags, but this does not make much of a difference here. We left it open, however, how to construct such probability spaces, and in particular how to define a suitable measurable spaces $(\mathbb{D}, \mathfrak{D})$, which is nontrivial for uncountable $\mathbb{D}$. In this section, we provide a general construction for constructing such measurable spaces.

3.1 Probabilistic Databases as Finite Point Processes

Throughout this paper, we only consider database schemas $\mathcal{S}$ where for every attribute A the domain $\mathsf{dom}_{\mathcal{S}}(A)$ is a Polish space. This is no real restriction; all domains one might typically find, such as the sets of integers, reals, or strings over a finite or even countable alphabet have this property.

In the following, we fix a database schema $\mathcal{S} = (\mathcal{A}, \mathcal{R})$. It follows from Fact 4 that not only the domains $\mathsf{dom}_{\mathcal{S}}(A)$ of the attributes $A \in \mathcal{A}$, but also the spaces $\mathsf{dom}_{\mathcal{S}}(R)$ and $\mathsf{facts}_{\mathcal{S}}(R)$ for all $R \in \mathcal{R}$ are Polish. We equip all of these spaces with their respective Borel σ-algebras and note that $\mathsf{dom}_{\mathcal{S}}(R)$ and $\mathsf{facts}_{\mathcal{S}}(R)$ are Borel-isomorphic from the point of view of measurable spaces. Thus, they can be used interchangeably when discussing measurability issues with respect to a single relation. For the set $\mathsf{facts}_{\mathcal{S}}(R)$ of facts *using relation symbol* $R \in \mathcal{R}$, let $\mathfrak{F}_{\mathcal{S}}(R)$ denote its (Borel) σ-algebra. We equip $\mathsf{facts}_{\mathcal{S}}(\mathcal{R})$, the set of *all facts of schema* $\mathcal{S}$ with the σ-algebra

$$\mathfrak{F}_{\mathcal{S}}(\mathcal{R}) = \{F \subseteq \mathsf{facts}_{\mathcal{S}}(\mathcal{R}) \colon F \cap \mathsf{facts}_{\mathcal{S}}(R) \in \mathfrak{F}_{\mathcal{S}}(R) \text{ for all } R \in \mathcal{R}\}.$$

Note that this is indeed a σ-algebra and, moreover, turns $(\mathsf{facts}_{\mathcal{S}}(\mathcal{R}), \mathfrak{F}_{\mathcal{S}}(\mathcal{R}))$ into a standard Borel space (cf. [30, p. 39] and [29, p. 166]).

Now a probabilistic database of schema $\mathcal{S}$ is supposed to be a probability space $(\mathbb{D}, \mathfrak{D}, P)$ where $\mathbb{D} \subseteq \mathbb{D}_{\mathcal{S}}$. Without loss of generality we may assume that actually $\mathbb{D} = \mathbb{D}_{\mathcal{S}} = \left(\!\!\binom{\mathsf{facts}_{\mathcal{S}}(\mathcal{R})}{<\omega}\!\!\right)$, because we can adjust the probability measure to be 0 on instances we are not interested in. Thus a probabilistic database is a probability space over finite sets of facts. This is exactly what a finite point process over the state space consisting of facts is. We still need to define the σ-algebra $\mathfrak{D}$, but the theory of point processes gives us a generic way of doing this: we let $\mathfrak{D}_{\mathcal{S}} = \mathfrak{C}_{\mathsf{facts}_{\mathcal{S}}(\mathcal{R})}$ be the counting σ-algebra of $\mathbb{D}_{\mathcal{S}}$ (cf. Section 2.3).

▶ **Definition 6.** *A* standard probabilistic database *of schema $\mathcal{S}$ is a probability space $(\mathbb{D}_{\mathcal{S}}, \mathfrak{D}_{\mathcal{S}}, P)$.*

That is, a standard probabilistic database of schema $\mathcal{S}$ is a finite point process over the state space $(\mathsf{facts}_{\mathcal{S}}(\mathcal{R}), \mathfrak{F}_{\mathcal{S}})$.

The reason we speak of "standard" PDBs in the definition is to distinguish them from the more general PDBs introduced in [38, Definition 3.1]. In [38], we left the σ-algebra unspecified and only required the (mild) property, that the occurrence of measurable sets of facts is themselves measurable. This requirement corresponds to a set version of the counting events defined above and is thus given by default in a standard probabilistic database.

Even though the construction of counting σ-algebras for point processes is nontrivial, we are convinced that it is a natural generic construction of σ-algebras over spaces of finite (or countable) sets and the extensive usage of these constructions throughout mathematics for more than fifty years now indicates their suitability for such tasks. *Throughout this paper, all probabilistic databases are standard. Therefore, we omit the qualifier "standard" in the following and just speak of probabilistic databases (PDBs).*

We defined instances of PDBs to be bags of facts. However, if a PDB, that is, a finite point process is simple (see Section 2.3), then it may be interpreted as a PDB with set-instances.

▶ **Example 7.** Every finite probabilistic database (as introduced, for example, in [63]) can be viewed as a standard PDB: Let $\tilde{\mathbb{D}}$ be a finite set of set-valued database instances over some schema $\mathcal{S} = (\mathcal{A}, \mathcal{R})$ and let $\tilde{P}\colon \tilde{\mathbb{D}} \to [0,1]$ a probability measure on $\tilde{\mathbb{D}}$ (equipped with the power set as its σ-algebra). Then $(\tilde{\mathbb{D}}, \tilde{P})$ corresponds to the simple finite point process $(\mathbb{D}, \mathfrak{D}, P)$ on the instance measurable space of $\mathcal{S}$ with state space $(\mathsf{facts}_{\mathcal{S}}(\mathcal{R}), \mathfrak{F}_{\mathcal{S}}(\mathcal{R}))$ where $P(\mathcal{D}) = \tilde{P}(\mathcal{D} \cap \tilde{\mathbb{D}})$ (interpreting $\tilde{\mathbb{D}}$ with a (finite) collection of bags with $\{0,1\}$-valued multiplicities).

3.2 The Possible Worlds Semantics of Queries and Views

In the traditional database setting, *views* are mappings from database instances of an *input schema* (or *source schema*) $\mathcal{S} = (\mathcal{A}, \mathcal{R})$ to database instances of some *output schema* (or *target schema*) $\mathcal{S}' = (\mathcal{A}', \mathcal{R}')$. Views, whose output schema $\mathcal{S}'$ consists of a single relational symbol only are called *queries*. Queries and views are usually given by syntactic expressions in some *query language*. As it is common, we will blur the distinction between a query (or view) and its syntactic representation.

Let $\Delta = (\mathbb{D}_{\mathcal{S}}, \mathfrak{D}_{\mathcal{S}}, P)$ be a probabilistic database of schema $\mathcal{S} = (\mathcal{A}, \mathcal{R})$ and let V be a view of input schema $\mathcal{S}$ and output schema $\mathcal{S}' = (\mathcal{A}', \mathcal{R}')$. The image of a set $\mathcal{D} \subseteq \mathfrak{D}$ of instances is $V(\mathcal{D}) = \{V(D)\colon D \in \mathcal{D}\} \subseteq \mathbb{D}_{\mathcal{S}'}$.

Now we would like to define a probability measure on the output space $(\mathbb{D}_{\mathcal{S}'}, \mathfrak{D}_{\mathcal{S}'})$ by

$$P'(\mathcal{D}') := P\big(V^{-1}(\mathcal{D}')\big) = P\big(\{D \in \mathbb{D}\colon V(D) \in \mathcal{D}'\}\big) \tag{1}$$

for all $\mathcal{D}' \in \mathfrak{D}_{\mathcal{S}'}$. Then V would map Δ to $\Delta' := (\mathbb{D}_{\mathcal{S}'}, \mathfrak{D}_{\mathcal{S}'}, P')$. This semantics of views over PDBs is known as the *possible worlds semantics* of probabilistic databases [36, 3, 63, 65].

However, P' (as defined in (1)) is only well-defined if for all $\mathcal{D}' \in \mathfrak{D}_{\mathcal{S}'}$ the set $V^{-1}(\mathcal{D}')$ is in $\mathfrak{D}_{\mathcal{S}}$, that is, if V is a *measurable* mapping from $(\mathbb{D}_{\mathcal{S}}, \mathfrak{D}_{\mathcal{S}})$ to $(\mathbb{D}_{\mathcal{S}'}, \mathfrak{D}_{\mathcal{S}'})$.

Measurability is not just a formality, but an issues that requires attention. The following example shows that there are relatively simple "queries" that are not measurable.

▶ **Example 8.** Let $\mathcal{S} = \mathcal{S}'$ be the schema consisting of a singe unary relation symbol R with attribute domian $\mathbb{R}$ (equipped with the Borel σ-algebra), and let B be some Borel set in $\mathbb{R}^2$.

We define a mapping $Q_B\colon \mathbb{D}_{\mathcal{S}} \to \mathbb{D}_{\mathcal{S}}$, our "query", by

$$Q_B(D) := \begin{cases} D & \text{if } D \text{ is a singleton } \{\!\{R(x)\}\!\} \text{ and there exists } \; y \in \mathbb{R} \text{ s.t. } (x,y) \in B, \\ \emptyset & \text{otherwise.} \end{cases}$$

Observe that $Q_B^{-1}(\mathbb{D}_{\mathcal{S}}) = \{\{\!\{R(x)\}\!\}\colon x \in \mathsf{proj}_1(B)\}$, where $\mathsf{proj}_1(B) = \{x \in \mathbb{R}\colon \text{there is } y \in \mathbb{R} \text{ s.t. } (x,y) \in B\}$. It is a well known fact that there are Borel sets $B \subseteq \mathbb{R}^2$ such that the projection $\mathsf{proj}_1(B)$ is *not* a Borel set in $\mathbb{R}$ (see [62, Theorem 4.1.5]). For such sets B, the query Q_B is not measurable.

The rest of this paper is devoted to proving that queries and views expressed in standard query languages, specifically relational algebra, possibly extended by aggregation, and Datalog queries, are measurable mappings and thus have a well-defined open-world semantics over probabilistic databases.

It will be sufficient to focus on *queries*, because views can be composed from queries and the measurability results can be lifted (as we formally show in the next subsection). *Throughout the rest of the paper, we adopt the following notational conventions: queries are denoted by Q and map a PDB $\Delta = (\mathbb{D}, \mathfrak{D}, P)$ to a PDB $\Delta' = (\mathbb{D}', \mathfrak{D}', P')$ such that Δ is of schema $\mathcal{S}$ and Δ' is of schema $\mathcal{S}'$.*

▶ Observation 9. The task of establishing measurability of queries in our framework is simplified by the following.

1. If we want to demonstrate the measurability of Q, it suffices to show that $Q^{-1}(\mathcal{D}') \in \mathfrak{D}$ for all *counting events* $\mathcal{D}' = \mathcal{C}'(F, n)$ of $(\mathbb{D}', \mathfrak{D}')$. This is due to Fact 2 because they generate $\mathfrak{D}'$.
2. Since compositions of measurable mappings are measurable (again from Fact 2), composite queries are immediately measurable if all their components are measurable queries to begin with. In particular, we can demonstrate the measurability of general queries of some query language by structural induction.

▶ Remark 10. Let us again mention something related to the well-established knowledge on point processes. The mappings (queries) we investigate map between point processes that are defined on different measure spaces that are themselves a conglomerate of simpler measure spaces of different shape. It is well-known that measurable transformations of the state space of a point process define a new point process on the transformed state space (a strengthening of this result is commonly referred to as the "mapping theorem" [48]). Our queries however are in general already defined on point configurations and not on the state space of facts. Thus, their measurability can in general not be obtained by the idea just sketched.

3.3 Assembling Views from Queries

We think of views as finite sets of queries, including one for every relation of the output schema. Suppose $V = \{Q_1, \dots, Q_k\}$ is a view consisting of measurable queries $Q_1, \dots, Q_k$ where the names of the target relations of the Q_i are mutually distinct. The target schema $\mathcal{S}'$ of V is given by the union of the target schemas of Vs individual queries. Now every fact $f \in \mathsf{facts}_{\mathcal{S}'}(\mathcal{R}')$ of the new schema originates from the target schema of exactly one of the queries $Q_1, \dots, Q_k$. We refer to that query as Q_f. Then for all $D \in \mathbb{D}$ and $f \in \mathsf{facts}_{\mathcal{S}'}(\mathcal{R}')$, we define $\#_{V(D)}(f) \coloneqq \#_{Q_f(D)}(f)$. Now if $F \subseteq \mathsf{facts}_{\mathcal{S}'}(\mathcal{R}')$, let $F_i \coloneqq F \cap \mathsf{facts}_{\mathcal{S}'_i}(\mathcal{R}'_i)$ where $\mathcal{S}'_i = (\mathcal{A}'_i, \mathcal{R}'_i)$ is the target schema of Q_i. Then

$$\#_{V(D)}(F) = n \quad \Leftrightarrow \quad \text{there are } n_1, \dots, n_k \text{ with } \textstyle\sum_{i=1}^{k} n_i = n \text{ such that } \#_{Q_i(D)}(F_i) = n_i.$$

Since the F_i are measurable if and only if F is measurable, the above describes a countable union of measurable sets. Thus, V is measurable.

4 Relational Algebra

As motivated in Section 3.2, we now investigate the measurability of relational algebra queries in our model. The concrete relational algebra for bags that we use here is basically the (unnested version of the) algebra that was introduced in [21] and investigated respectively extended and surveyed in [5, 40, 39]. It is called BALG^1 (with superscript 1) in [40]. We do not introduce nesting as it would yield yet another layer of abstraction and complexity to the spaces we investigate, although by the properties that such spaces exhibit, we have strong reason to believe that there is no technical obstruction in allowing spaces of finite bags as attribute domains.

The operations we consider are shown in the Table 1 below. As seen in [5, 40, 39], there is some redundancy within this set of operations that will be addressed later. A particular motivation for choosing this particular algebra is that possible worlds semantics are usually built on top of set semantics and these operations naturally extend the common behavior of relation algebra queries to bags. This is quite similar to the original motivation of [21] and [5] regarding their choice of operations.

Table 1 BALG^1-operators considered in this paper.

Base Queries	Constructors	$Q = \{\!\{\}\!\}$ and $Q = \{\!\{R(a)\}\!\}$
	Extractors	$Q = R$
	Renaming	$Q = \varrho_{A\to B}(R)$
Basic Bag Operations	Additive Union	$Q = R_1 \uplus R_2$
	Difference	$Q = R_1 - R_2$
	Max-Union	$Q = R_1 \cup R_2$
	(Min-)Intersection	$Q = R_1 \cap R_2$
	Deduplication	$Q = \delta(R)$
SPJ-Operations	Selection	$Q = \sigma_{(A_1,\dots,A_k)\in B}(R)$
	Projection	$Q = \pi_{(A_1,\dots,A_k)}(R)$
	Cross Product	$Q = R_1 \times R_2$

The main result we establish in this section is the following theorem:

▶ **Theorem 11.** *All queries expressible in the bag algebra* BALG^1 *are measurable.*

Since compositions of measurable mappings are measurable, the measurability of the operators from Table 1 directly entails the measurability of compound queries by structural induction.

First note that the measurability of the base queries is easy to prove.

▶ **Lemma 12.** *The queries* $\{\!\{\}\!\}$, $\{\!\{R(a)\}\!\}$ *and* R *are measurable.*

Proof. First consider $Q = \{\!\{\}\!\}$ and fix some $\mathcal{D}' \in \mathfrak{D}'$. If $\{\!\{\}\!\} \in \mathcal{D}'$, then $Q^{-1}(\mathcal{D}') = \mathbb{D} \in \mathfrak{D}$. Otherwise, $Q^{-1}(\mathcal{D}') = \emptyset \in \mathfrak{D}$. Thus, Q is measurable. The same argument applies to $Q = \{\!\{R(a)\}\!\}$.

Now consider the query $Q = R$ and let $\mathcal{C}'(F, n)$ be a counting event in the output measurable space. Then for every instance $D \in \mathbb{D}$, $\#_{Q(D)}(F) = n$ if and only if $\#_D(F) = n$ Thus, $Q^{-1}(\mathcal{C}'(F, n))$ is the counting event $\mathcal{C}(F, n)$ in $(\mathbb{D}, \mathfrak{D})$. Hence, Q is measurable. ◀

4.1 Basic Bag Operations

We will obtain the measurability of the basic bag operations $\uplus$, $-$, $\cap$, $\cup$, δ as a consequence of the following, more general result that gives some additional insight into properties that make queries measurable.

Consider a query Q of input schema $\mathcal{S}$ and output schema $\mathcal{S}'$ operating on relations R_1 and R_2 of $\mathcal{S}$. Let R' be the single (output) relation of $\mathcal{S}'$.

▶ **Lemma 13.** *Suppose that given Q there exist functions $q_1 \colon \mathsf{facts}_{\mathcal{S}'}(R') \to \mathsf{facts}_{\mathcal{S}}(R_1)$ and $q_2 \colon \mathsf{facts}_{\mathcal{S}'}(R') \to \mathsf{facts}_{\mathcal{S}}(R_2)$ with the following properties:*

1. *for all $n \in \mathbb{N}$ there exists a set $M(n) \subseteq \mathbb{N}^2$ with $(0,0) \notin M(n)$ for $n > 0$ such that for all $D \in \mathbb{D}$ and all $f \in \mathsf{facts}_{\mathcal{S}'}(R')$ it holds that*

$$\#_{Q(D)}(f) = n \qquad \textit{if and only if} \qquad \big(\#_D(q_1(f)), \#_D(q_2(f))\big) \in M(n);$$

2. *both q_1 and q_2 are injective and continuous;*
3. *the images of F under q_1 and q_2 are measurable: $q_1(F) \in \mathfrak{F}_{\mathcal{S}}(R_1)$ and $q_2(F) \in \mathfrak{F}_{\mathcal{S}}(R_2)$.*

Then Q is measurable.

Let us briefly mention the impact of the various preconditions of the lemma before turning to its proof. The existence of the functions q_1 and q_2 ensures that preimages of counting events $\mathcal{C}'(F, n)$ under the query Q can be approximated by using the fact that our state spaces are Polish. They "decompose" the set F of facts into disjoint (and measurable!) sets of facts for the preimage in a continuous, invertible way that exactly captures how tuples in the preimage relate to tuples in the image.

Proof (Lemma 13). Assume that q_1 and q_2 exist with properties 1 to 3. We fix $F \in \mathfrak{F}_{\mathcal{S}'}(R')$ and $n \in \mathbb{N}_+$ and show that $Q^{-1}(\mathcal{C}'(F, n))$ is in $\mathfrak{D}$. Let F_0 be a countable, dense set in $\mathsf{facts}_{\mathcal{S}'}(R')$. We claim that $\#_{Q(D)}(F) = n$ if and only if

$$\begin{aligned}
&\text{there exist } \ell \in \mathbb{N}_+ \text{ and } n_1, \dots, n_\ell \in \mathbb{N} \text{ with } \textstyle\sum_{i=1}^{\ell} n_i = n \text{ and}\\
&\quad\text{there exist } (n_{i,1}, n_{i,2}) \in M(n_i) \text{ and } k_0 \in \mathbb{N}_+ \text{ and}\\
&\qquad\text{there exist Cauchy sequences } (f_1^k)_{k\in\mathbb{N}}, \dots, (f_\ell^k)_{k\in\mathbb{N}} \text{ in } F_0 \text{ with}\\
&\qquad B_{1/k_0}(f_i^k) \cap B_{1/k_0}(f_{i'}^{k'}) = \emptyset \text{ for all } k, k' \text{ and } i \neq i' \text{ such that for all } k > k_0 \qquad (*)\\
&\qquad\quad \#_D(q_1(F) \cap B_{1/k}(q_1(f_i^k))) = n_{i,1} \text{ and } \#_D(q_2(F) \cap B_{1/k}(q_2(f_i^k))) = n_{i,2}\\
&\qquad\qquad \text{for } 1 \leq i \leq \ell \text{ and}\\
&\qquad\quad \#_D(q_1(F) \setminus \textstyle\bigcup_{i=1}^{\ell} B_{1/k}(q_1(f_i^k))) = 0 \text{ and } \#_D(q_2(F) \setminus \textstyle\bigcup_{i=1}^{\ell} B_{1/k}(q_2(f_i^k))) = 0.
\end{aligned}$$

Note that $(*)$ is a countable combination of counting events in $(\mathbb{D}, \mathfrak{D})$ (using condition 3, in particular). Thus, to show the measurability of Q it suffices to show the equivalence of $\#_{Q(D)}(F) = n$ and $(*)$.

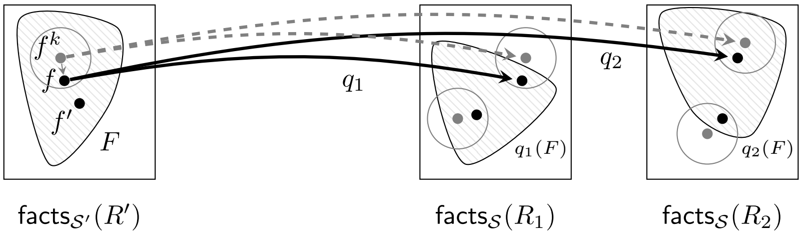
f^k
f
f'
F
q_1
q_2
$q_1(F)$
$q_2(F)$
$\mathsf{facts}_{\mathcal{S}'}(R')$
$\mathsf{facts}_{\mathcal{S}}(R_1)$
$\mathsf{facts}_{\mathcal{S}}(R_2)$

Figure 1 Example illustration of $(*)$ for two facts f and f'. Both these facts are approximated by Cauchy sequences that under q_1 and q_2 also approximate their images.

Assume $\#_{Q(D)}(F) = n$. Let $f_1, \dots, f_\ell$ be the facts from F with the property that $\#_D(q_1(f)) > 0$ or $\#_D(q_2(f)) > 0$.

Let $n_i := \#_{Q(D)}(f_i)$. From condition 1 we know that $(\#_D(q_1(f_i)), \#_D(q_2(f_i))) \in M(n_i)$ as well as $\sum_{i=1}^{\ell} n_i = n$. Let $(f_1^k), \dots, (f_\ell^k)$ be Cauchy sequences from F_0 that converge to $f_1, \dots, f_\ell$. Since ℓ is finite, the balls around f_i^k and $f_{i'}^k$ do not intersect for sufficiently large k as well as the balls around their images under q_1 respectively q_2 (since both of them are injective and continuous). Thus, $\#_D(q_1(F) \cap B_{1/k}(q_1(f_i^k))) = \#_D(q_1(f_i))$ and $\#_D(q_2(F) \cap B_{1/k}(q_2(f_i^k))) = \#_D(q_2(f_i))$ for sufficiently large k. Therefore, D satisfies $(*)$.

Now for the other direction, suppose D satisfies $(*)$. As the f_i^k are Cauchy sequences, the spaces $\mathsf{facts}_{\mathcal{S}'}(R_j)$ are Polish and hence complete, and the q_j are continuous there exists (for every $1 \leq i \leq \ell$) some $f_i \in F$ such that $f_i^k \to f_i$, $q_1(f_i^k) \to q_1(f_i)$ and $q_2(f_i^k) \to q_2(f_i)$ as

$k \to \infty$ and $(\#_D(q_1(f_i)), \#_D(q_2(f_i))) = (n_{i,1}, n_{i,2}) \in M(n_i)$. By condition 1, $Q(D)$ contains f_i with multiplicity n_i and as $\sum_{i=1}^{\ell} n_i = n$ (and since D had no other facts with positive multiplicity than the above), it follows that $\#_{Q(D)}(F) = n$. ◀

Note that the result above easily generalizes to queries that depend on an arbitrary number of relations of the input probabilistic database. Lemma 13 provides a criterion to establish the measurability of queries. Checking its precondition for bag operations we consider turns out to be quite easy and yields the following lemma.

▶ **Lemma 14.** *The following queries are measurable:*

1. (Additive Union) $Q = R_1 \uplus R_2$ *with* $R_1, R_2 \in \mathcal{R}$ *of equal type.*
2. (Difference) $Q = R_1 - R_2$ *with* $R_1, R_2 \in \mathcal{R}$ *of equal type.*
3. ((Min-)Intersection) $Q = R_1 \cap R_2$ *with* $R_1, R_2 \in \mathcal{R}$ *of equal type.*
4. (Max-Union) $Q = R_1 \cup R_2$ *with* $R_1, R_2 \in \mathcal{R}$ *of equal type.*
5. (Deduplication) $Q = \delta(R)$ *with* $R \in \mathcal{R}$.

Proof. As $\cup$ and $\cap$ are expressible via $\uplus$ and $-$ (cf. [5]), we only show Statements 1, 2 and 5.

1. Define q_1 and q_2 by $q_i(R(x)) = R_i(x)$. Then q_i, $i \in \{1, 2\}$ is injective and continuous and $q_i(F) = F_i \in \mathfrak{F}_{\mathcal{S}}(R_i)$. Now let $k \in \mathbb{N}$ and let $M(k) \subseteq \mathbb{N}^2$ be the set of pairs (k_1, k_2) with the property that $k_1 + k_2 = k$. Then $\#_{Q(D)}(f) = k$ if and only if $\big(\#_D\big(q_1(f)\big), \#_D\big(q_2(f)\big)\big) \in M(k)$. Together, by Lemma 13, Q is measurable.
2. This works exactly like in the case of $\uplus$ with $M(k)$ being the set of pairs (k_1, k_2) with $\mathsf{max}(k_1 - k_2, 0) = k$.
5. In this case, we only use a single function q that maps $R'(x)$ to $R(x)$. Again, q is obviously both continuous and injective and $q(F) \in \mathfrak{F}_{\mathcal{S}}(R)$ for every measurable F. If $k = 1$, we let $M(k) = \mathbb{N} \setminus \{0\}$ and $M(k) = \{0\}$ otherwise. Then clearly $\#_{Q(D)}(R(x)) = k$ if and only if $\#_D(q(R(x)) \in M(k)$ and again, Q is measurable by Lemma 13. ◀

4.2 Selection, Projection and Join

In this section, we investigate selection and projection as well as the cross product of two relations. We start with the following helpful lemma that allows us to restructure our relations into a more convenient shape to work with. Semantically, it might be seen as a special case of a projection query.

▶ **Lemma 15.** *Reordering attributes within the type of a relation yields a measurable query.*

Proof. Recall that any permutation can be expressed as a composition of transpositions. Thus, we only consider the case where two attributes, say A and B, switch places within the type of some relation $R \in \mathcal{R}$. Let q be the function that maps $\mathsf{facts}_{\mathcal{S}}(R)$ to $\mathsf{facts}_{\mathcal{S}'}(R')$ by swapping the entries for attribute A and B. Obviously, under q, the preimage of a measurable rectangle in $\mathfrak{F}_{\mathcal{S}'}(R)$ is a measurable rectangle itself. As $\#_{Q(D)}(F) = n$ if and only if $\#_D(q^{-1}(F)) = n$, Q is measurable. ◀

▶ **Lemma 16.** *The query* $Q = \sigma_{(A_1,\dots,A_k)\in\mathcal{B}}(R)$ *is measurable for all* $R \in \mathcal{R}$*, all pairwise distinct attributes* $A_1, \dots, A_k \in \mathsf{type}_{\mathcal{S}}(R)$ *and all Borel subsets* $\mathcal{B}$ *of* $\prod_{i=1}^{k} \mathsf{dom}_{\mathcal{S}}(A_i)$.

Proof. Fix some $F \in \mathfrak{F}_{\mathcal{S}'}(R')$ and $n \in \mathbb{N}$. By Lemma 15, we may assume that $\mathsf{type}_{\mathcal{S}}(R) = (A_1, \dots, A_m)$ where $m \geq k$. Let $F_{\mathcal{B}} := \{R\} \times \mathcal{B} \times \mathsf{dom}_{\mathcal{S}}(A_{k+1}) \times \dots \times \mathsf{dom}_{\mathcal{S}}(A_m)$. Note that $F_{\mathcal{B}} \in \mathfrak{F}_{\mathcal{S}}(R)$. (This is a consequence of Fact 4.) As $\#_{Q(D)}(F) = n$ if and only if $n = \#_{Q(D)}(F \cap F_{\mathcal{B}}) = \#_D(F \cap F_{\mathcal{B}})$, Q is measurable. ◀

▶ **Example 17.** Assume that $\mathsf{dom}_{\mathcal{S}}(A) = \mathsf{dom}_{\mathcal{S}}(B) = \mathbb{R}$ and both A and B appear in the type of $R \in \mathcal{R}$. It is well-known (and can be shown by standard arguments) that the sets $\mathcal{B}_{=} \coloneqq \{(x,y) \in \mathbb{R}^2 \colon x = y\}$ and $\mathcal{B}_{<} \coloneqq \{(x,y) \in \mathbb{R}^2 \colon x < y\}$ are Borel in $\mathbb{R}^2$. Thus $\sigma_{A=B}(R) \coloneqq \sigma_{(A,B)\in\mathcal{B}_{=}}(R)$ and $\sigma_{A<B}(R) \coloneqq \sigma_{(A,B)\in\mathcal{B}_{<}}(R)$ are measurable by Lemma 16.

▶ **Lemma 18.** . *The query $Q = \pi_{A_1,\dots,A_k}(R)$ is measurable for all $R \in \mathcal{R}$ and all mutual distinct $A_1,\dots,A_k \in \mathsf{type}_{\mathcal{S}}(R)$.*

Proof. Again, fix some $F \in \mathfrak{F}_{\mathcal{S}'}(R')$ and $n \in \mathbb{N}$. Note that F is of the shape $\{R'\} \times \mathcal{B}$ where $\mathcal{B}$ is Borel in $\mathsf{dom}_{\mathcal{S}'}(R') = \prod_{i=1}^{k} \mathsf{dom}_{\mathcal{S}}(A_k)$. By Lemma 15, we may again assume that $\mathsf{type}_{\mathcal{S}}(R) = (A_1,\dots,A_m)$ with $m \geq k$. Define $F_{\mathcal{B}}$ exactly like in the proof of Lemma 16: $F_{\mathcal{B}} \coloneqq \{R\} \times \mathcal{B} \times \mathsf{dom}_{\mathcal{S}}(A_{k+1}) \times \dots \times \mathsf{dom}_{\mathcal{S}}(A_m)$. Again, $F_{\mathcal{B}} \in \mathfrak{F}_{\mathcal{S}}(R)$. Now, we have $\#_{Q(D)}(F) = n$ if and only if $\#_D(F_{\mathcal{B}}) = n$ and hence, Q is measurable. ◀

▶ **Lemma 19.** *The query $Q = R_1 \times R_2$ is measurable for all $R_1, R_2 \in \mathcal{R}$.*

First we note that this turns out to be more involved than it seems on first sight. The straight-forward approach would be to take a counting event $\mathcal{C}(F,n)$ in the output measurable space and to decompose F into its “left and right parts” $F_1 \subseteq \mathsf{facts}_{\mathcal{S}}(R_1)$ and $F_2 \subseteq \mathsf{facts}_{\mathcal{S}}(R_2)$ such that the instances from the preimage of the query are exactly those with $\#_D(F_1) = n_1$ and $\#_D(F_2) = n_2$ such that $n_1 \cdot n_2 = n$, similar to the setting of Lemma 13. This approach does not settle the case since the sets F_1 and F_2 need not be measurable in general (see [62, Theorem 4.1.5]; we used the same argument in Example 8) which in particular violates the second precondition of Lemma 13.

Proof Sketch. Using renaming, we may assume that the types of R_1 and R_2 are disjoint in terms of attribute names. Consider $F \in \mathfrak{F}_{\mathcal{S}'}(R')$ and $n \in \mathbb{N}$. If F is a measurable rectangle $F = F_1 \times F_2$, it is easy to see that the naïve approach sketched above works via $\#_{Q(D)}(F) = n$ if and only if $\#_D(F_1) \cdot \#_D(F_2) = n$.

In the general case of F being an arbitrary Borel set, we consider the k-*coarse* preimage of $\mathcal{C}'(F,n)$ first. These are the database instances from $\mathbb{D}$ whose minimal inter-tuple distance is at least $\frac{1}{k}$ for some fixed Polish metrics. One can show that these k-coarse preimages of the query are measurable for all F, n and k. As the union of these preimages over all positive integers k is exactly the preimage of $\mathcal{C}'(F,n)$, Q is measurable. The details of this proof can be found in the full version of the paper [37]. ◀

Altogether, within the last three sections, we have established the measurability of all the (bag) relational algebra operators from Table 1 and thus have proven Theorem 11. Of course any additional operator that is expressible by a combination of operations from Table 1 is immediately measurable as well, including for example *natural joins* $Q = R_1 \bowtie R_2$ or selections where the selection predicate is a Boolean combinations of predicates of the shape $(A_1,\dots,A_k) \in \mathcal{B}$.

5 Aggregate Queries

In this section, we study various kinds of aggregate operators. Let U and V be standard Borel spaces. An *aggregate operator* (or *aggregator*) from U to V is a mapping Φ that sends bags of elements of U to elements of V: $\Phi \colon \left(\!\!\binom{U}{<\omega}\!\!\right) \to V$. Every such aggregator Φ gives rise to a query $Q = \varpi_{\Phi}(R)$ defined by $Q(D) \coloneqq \{\!\{R'(v)\}\!\}$ for $v \coloneqq \Phi(\{\!\{u \colon R(u) \in D\}\!\})$. (The notation we use for aggregation queries is loosely based on that of [28].) Observe that for every instance D, $\#_{Q(D)}\big(R'(v)\big) = 1$ if and only if $\Phi(\{\!\{u \colon R(u) \in D\}\!\}) = v$ (and 0 otherwise). It is easy to see that $Q = \varpi_{\Phi}(R)$ is a measurable query whenever Φ is measurable w. r. t. the counting σ-algebra on $\left(\!\!\binom{U}{<\omega}\!\!\right)$: we have $\#_{Q(D)}(F) = 1$ if and only if $D \in \{R\} \times \Phi^{-1}(\{v\colon R'(v) \in F\})$ (and $\#_{Q(D)}(F) = 0$ otherwise).

▶ **Example 20.** The following are the most common aggregate operators:

- (Count) $\mathsf{CNT}(\{\!\{a_1, \dots, a_n\}\!\}) = n$ and $\mathsf{CNTd}(\{\!\{a_1, \dots, a_n\}\!\}) = |\{a_1, \dots, a_n\}|$.
- (Sum) $\mathsf{SUM}(\{\!\{a_1, \dots, a_n\}\!\}) = a_1 + \dots + a_n$ where a_i are (for instance) real numbers.
- (Minimum / Maximum) $\mathsf{MIN}(\{\!\{a_1, \dots, a_n\}\!\}) = \min\{a_1, \dots, a_n\}$ and $\mathsf{MAX}(\{\!\{a_1, \dots, a_n\}\!\}) = \max\{a_1, \dots, a_n\}$ for ordered domains.
- (Average) $\mathsf{AVG}(\{\!\{a_1, \dots, a_n\}\!\}) = \frac{1}{n}\big(a_1 + \dots + a_n\big)$ where the a_i might again be real numbers.

Note that ϖ_{CNT} and ϖ_{CNTd} are trivially measurable within our framework by the usage of the counting σ-algebra (and the measurability of deduplication for CNTd).

▶ **Lemma 21.** *For all $m \in \mathbb{N}$, let $\varphi_m\colon U^m \to V$ be a* symmetric *function, i. e., $\varphi_m(u) = \varphi_m(u')$ for all $u \in U^m$ and all permutations u' of u. If φ_m is measurable for all m, then $\Phi\colon \left(\!\!\binom{U}{<\omega}\!\!\right) \to V$ defined via $\Phi(\{\!\{u_1, \dots, u_m\}\!\}) \coloneqq \varphi_m(u_1, \dots, u_m)$ is measurable w. r. t. the counting σ-algebra on $\left(\!\!\binom{U}{<\omega}\!\!\right)$.*

Proof. It suffices to show that the restriction Φ_m of Φ to $\left(\!\!\binom{U}{m}\!\!\right)$ is measurable for all $m \in \mathbb{N}$. If $\mathcal{V}$ is Borel in V, then $\varphi_m^{-1}(\mathcal{V})$ is Borel in U^m as φ_m is measurable. Moreover, since φ_m is symmetric, $\varphi_m^{-1}(\mathcal{V})$ is a symmetric set (i. e. if $\bar{u} \in \varphi_m^{-1}(\mathcal{V})$, then every permutation of u is in $\varphi_m^{-1}(\mathcal{V})$ as well). But then $\Phi_m^{-1}(\mathcal{V})$ is measurable since there is a one-to-one correspondence between the measurable sets of $\left(\!\!\binom{U}{m}\!\!\right)$ and the symmetric Borel sets of U^m [49, Theorem 1]. ◀

As an example application of this lemma we note that all the mappings Φ that were introduced in Example 20 are measurable – the related mappings φ_m of Lemma 21 are all continuous and thus measurable in all of the cases.

A concept closely tied to aggregation is *grouping*. Suppose we want to group a relation R by its attributes $A_1, \dots, A_k$ and perform the aggregation only over the values of attribute A, and separately for every distinct $(A_1, \dots, A_k)$-entry in R. Without loss of generality, we assume that the type of R is $A_1 \times \dots \times A_k \times A$. We define a query $Q = \varpi_{A_1, \dots, A_k, \Phi(A)}(R)$ by

$$Q(D) = \{\!\{R'(\bar{u}, v)\colon R(\bar{u}) \in \pi_{A_1, \dots, A_k}(R(D)) \text{ and } v = \Phi(\{\!\{u\colon R(\bar{u}, u) \in D\}\!\})\}\!\}.$$

▶ **Lemma 22.** *Let $\mathsf{type}_{\mathcal{S}}(R) = A_1 \times \dots \times A_k \times A$ and $U = \mathsf{dom}_{\mathcal{S}}(A)$. If $\Phi\colon \left(\!\!\binom{U}{<\omega}\!\!\right) \to V$ is measurable (with U and V standard Borel), then $\varpi_{A_1, \dots, A_k, \Phi(A)}(R)$ is a measurable query.*

Proof. Let $Q = \varpi_{A_1, \dots, A_k, \Phi(A)}(R)$ and $\bar{A} = (A_1, \dots, A_k)$. Observe that for every tuple $x_1, \dots, x_n, \varepsilon$ with $x_i \in \prod_{j=1}^{k} \mathsf{dom}_{\mathcal{S}}(A_j)$ and $\varepsilon > 0$, the following query is a composition of measurable queries and thus measurable itself:

$$\tilde{Q}_{(x_1, \dots, x_n, \varepsilon)} = \bigcup_{i=1}^{n} \pi_{\bar{A}}\big(\sigma_{\bar{A} \in B_\varepsilon(x_i)}(R)\big) \times \varpi_{\Phi}\big(\pi_A\big(\sigma_{\bar{A} \in B_\varepsilon(x_i)}(R)\big)\big).$$

We have $\#_{Q(D)}(F) = n$ if and only if there exist pairwise distinct $f_1, \dots, f_n \in F$ such that $Q(D)$ has 1 hit in each of the f_i and nowhere else in F. Having D fixed, every f_i determines the value of the $(A_1, \dots, A_k)$-part of an R-fact in D. Call this tuple y_i. We can fix a countable sequence of $(n+1)$-tuples $(x_1, \dots, x_n, \varepsilon)$ such that (1) all x_i are from a countable dense set in $\prod_{j=1}^{k} \mathsf{dom}_{\mathcal{S}}(A_j)$, (2) $d(x_i, y_i) < \varepsilon$ for some fixed Polish metric, and, (3) $\varepsilon \to 0$. Then Q is the (pointwise) limit of the $\tilde{Q}_{(x_1, \dots, x_n, \varepsilon)}$ and, as such, Q is measurable. ◀

As noted before, the aggregates of Example 20 easily satisfy the precondition of Lemma 22.

▶ **Corollary 23.** *The query $\varpi_{A_1, \dots, A_k, \Phi}(R)$ with $A_1, \dots, A_k \in \mathsf{type}_{\mathcal{S}}(R)$ is measurable for all aggregates $\Phi \in \{\mathsf{CNT}, \mathsf{CNTd}, \mathsf{SUM}, \mathsf{MIN}, \mathsf{MAX}, \mathsf{AVG}\}$.*

6 Datalog Queries

In this section, we want to show that our measurability results extend to Datalog queries and in fact all types of queries with operators based on countable iterative (or inductive, inflationary, fixed-point) processes. We will not introduce Datalog or any of the related query languages. The details in the definitions do not matter when it comes to measurability of the queries. Here, we only consider set PDBs and queries with a set (rather than bag) semantics. The key observation is the following lemma.

▶ **Lemma 24.** *Let Q_i, for $i \in \mathbb{N}_+$, be a countable family of measurable queries of the same schema such that $Q = \bigcup_{i\geq 1} Q_i$, defined by $Q(D) := \bigcup_{i\geq 0} Q_i(D)$ for every instance D, is a well-defined query (that is, $Q(D)$ is finite for every D). Then Q is measurable.*

Proof. For every $n \in \mathbb{N}_+$, let $Q^{(n)} := \bigcup_{i=1}^{n} Q_i$. As a finite union of measurable queries, $Q^{(n)}$ is measurable. Since $Q = \lim_{n\to\infty} Q^{(n)}$, the measurability of Q follows. ◀

As every Datalog query can be written as a countable union of conjunctive queries, we obtain the following corollary.

▶ **Corollary 25.** *Every Datalog query is measurable.*

The same is true for queries in languages like inflationary Datalog or least fixed-point logic. For partial Datalog / fixed-point logic, we cannot directly use Lemma 24, but a slightly more complicated argument still based on countable limits works there as well.

7 Beyond Possible Worlds Semantics

In the literature on probabilistic databases, and motivated by real world application scenarios, also other kinds of queries have been investigated that have no intuitive description in the possible worlds semantics framework. A range of such queries is surveyed in [3, 67]. The reason for the poor integration into possible worlds semantics is because such queries lack a sensible interpretation on single instances that could be lifted to PDB events. Instead, they directly refer to the probability space of all instances.

Notable examples of such queries (cf. [47, 3, 67]) are:

- *probabilistic threshold queries* that intuitively return a deterministic table containing only those facts which have a marginal probability over some specified threshold;
- *probabilistic top-k-queries* that intuitively return a deterministic table containing the k most probable facts;
- *probabilistic skyline queries* [55] that consider how different instances compare to each other with respect to some notion of *dominance*; and
- *conditioning* [47] the probabilistic database to some event.

Note that the way we informally explained the first two queries above is only sensible if the space of facts is discrete. In a continuous setting, we interpret these queries with respect to a suitable countable partition of the fact space into measurable sets.

Let $\Delta_{\mathcal{S}}$ denote the class of probabilistic databases of schema $\mathcal{S}$. Note that all PDBs in $\Delta_{\mathcal{S}}$ have the same instance measurable space $(\mathbb{D}, \mathfrak{D})$. Queries and, more generally, views of input schema $\mathcal{S}$ and output schema $\mathcal{S}'$ are now mappings $V \colon \Delta_{\mathcal{S}} \to \Delta_{\mathcal{S}'}$.

We classify views in the following way:

▶ **Definition 26.** *Let $V\colon \Delta_{\mathcal{S}} \to \Delta_{\mathcal{S}'}$ with $V\colon \Delta = (\mathbb{D}, \mathfrak{D}, P) \mapsto (\mathbb{D}', \mathfrak{D}', P') = \Delta'$.*

1. *Every view V is of* type I*.*
2. *The view V is of* type II *(or,* pointwise local*) if for every $\Delta \in \Delta_{\mathcal{S}}$ there exists a measurable mapping $q_\Delta \colon \mathbb{D} \to \mathbb{D}$ such that $P'(\mathcal{D}') = P(q_\Delta^{-1}(\mathcal{D}'))$ for every $\mathcal{D}' \in \mathfrak{D}$.*
3. *The view V is of* type III *(or,* uniformly local*) if there exists a measurable mapping $q\colon \mathbb{D} \to \mathbb{D}$ such that $P'(\mathcal{D}') = P(q^{-1}(\mathcal{D}'))$ for every $\mathcal{D}' \in \mathfrak{D}'$.*

Letting $\mathcal{V}^{\mathrm{I}}$, $\mathcal{V}^{\mathrm{II}}$ and $\mathcal{V}^{\mathrm{III}}$ denote the classes of type I, type II and type III views (from $\Delta_{\mathcal{S}}$ to $\Delta_{\mathcal{S}'}$). Then $\mathcal{V}^{\mathrm{III}}$ captures the possible worlds semantics of views. Obviously, $\mathcal{V}^{\mathrm{III}} \subseteq \mathcal{V}^{\mathrm{II}} \subseteq \mathcal{V}^{\mathrm{I}}$. The following examples show that these inclusions are strict.

▶ **Example 27.** Consider the query $Q = Q_\alpha(D) = \{f \in \mathsf{facts}_{\mathcal{S}}(R) \colon P(\mathcal{C}(f, > 0)) \geq \alpha\} = q_\Delta$ for some $\alpha > 0$. Note that the set of facts of marginal probability at least α is finite in every PDB [38], hence the query is well-defined. This query is of type II. However, considering the simple PDBs Δ_1 and Δ_2 and two distinct facts f and f' such that

- the only possible world of positive probability in Δ_1 is $\{\!\{f\}\!\}$ with $P_{\Delta_1}(\{\!\{f\}\!\}) = 1$;
- similarly, Δ_2 has the worlds $\{\!\{f\}\!\}$ and $\{\!\{f'\}\!\}$ with $P_{\Delta_2}(\{\!\{f\}\!\}) = P_{\Delta_2}(\{\!\{f'\}\!\}) = \frac{1}{2}$.

Suppose q exists like in the Definition 26, part 3 and consider the event $\mathcal{D}'$ that f' occurs (this is a set of instances in the target measurable space of Q_α). Then $P_{\Delta_1}(q^{-1}(\mathcal{D}')) = 0$ entails $\{\!\{f\}\!\} \notin q^{-1}(\mathcal{D}')$. On the other hand $P_{\Delta_2}(q^{-1}(\mathcal{D}')) = 1$ and thus $\{\!\{f\}\!\}, \{\!\{f'\}\!\} \in q^{-1}(\mathcal{D}')$, a contradiction. Thus, Q is type II, but not type III.

▶ **Example 28.** Fix some PDB Δ with three possible worlds D_1, D_2 and D_3 with probabilities $p_1 = \frac{1}{6}$, $p_2 = \frac{1}{3}$ and $p_3 = \frac{1}{2}$. Now consider the query Q that conditions Δ on the event $\{D_1, D_2\}$ and pick the database instance $D = D_1$. Then $P(D \cap \{D_1, D_2\}) = P(\{D_1\}) = \frac{1}{6}$ and $P(\{D_1, D_2\}) = \frac{1}{6} + \frac{1}{2} = \frac{4}{6}$. Thus, $P(Q^{-1}(D)) = \frac{1}{6} / \frac{4}{6} = \frac{1}{4}$, but there is no event $\mathcal{D}$ in Δ with the property that $P(\mathcal{D}) = 1/4$. Thus, Q is type I, but not type II.

8 Conclusions

In this work, we described how to construct suitable probability spaces for infinite probabilistic databases, completing the picture of [38]. The viability of this model as a general framework for finite *and infinite* databases is supported by its compositionality with respect to typical database queries. Our main technical results establish that standard query languages have a well-defined open-world semantics.

It might be interesting to explore, whether more in-depth results on point processes have a natural interpretation when it comes to probabilistic databases. We believe for example that there is a strong connection between the infinite independence assumptions that were introduced in [38] and the class of Poisson point processes (cf. [48, p. 52]).

In the last section of the paper, we briefly discussed queries for PDBs that go beyond the possible worlds semantics. Such queries are very relevant for PDBs and deserve a systematic treatment in their own right in an infinite setting.

References

1 Serge Abiteboul, T.-H. Hubert Chan, Evgeny Kharlamov, Werner Nutt, and Pierre Senellart. Capturing Continuous Data and Answering Aggregate Queries in Probabilistic XML. *ACM Transactions on Database Systems (TODS)*, 36(4):25:1–25:45, 2011. doi:10.1145/1804669.1804679.

2 Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, Boston, MA, USA, 1st edition, 1995.

3 Charu C. Aggarwal and Philip S. Yu. A Survey of Uncertain Data Algorithms and Applications. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 21(5):609–623, 2009. `doi:10.1109/TKDE.2008.190`.

4 Parag Agrawal and Jennifer Widom. Continuous Uncertainty in Trio. In *Proceedings of the 3rd VLDB Workshop on Management of Uncertain Data (MUD '09)*, pages 17–32, Enschede, The Netherlands, 2009. Centre for Telematics and Information Technology (CTIT).

5 Joseph Albert. Algebraic Properties of Bag Data Types. In *Proceedings of the 17th International Conference on Very Large Databases (VLDB 1991)*, pages 211–219, San Francisco, CA, USA, 1991. Morgan Kaufmann Publishers Inc.

6 Adrian Baddeley. Spatial Point Processes and Their Applications. In Wolfgang Weil, editor, *Stochastic Geometry*, Lecture Notes in Mathematics, chapter 1, pages 1–75. Springer, Berlin, Heidelberg, Germany, 1st edition, 2007.

7 Vince Bárány, Balder Ten Cate, Benny Kimelfeld, Dan Olteanu, and Zografoula Vagena. Declarative Probabilistic Programming with Datalog. *ACM Transactions on Database Systems (TODS)*, 42(4), 2017.

8 Daniel Barbará, Héctor García-Molina, and Daryl Porter. The Management of Probabilistic Data. *IEEE Transactions on Knowledge and Data Engineering*, 4(5):487–502, 1992. `doi:10.1109/69.166990`.

9 Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz. Ontology-Mediated Queries for Probabilistic Databases. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI '17)*, pages 1063–1069, Palo Alto, CA, USA, 2017. AAAI Press.

10 Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz. Recent Advances in Querying Probabilistic Knowledge Bases. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI '18)*, pages 5420–5426. International Joint Conferences on Artificial Intelligence, 2018. `doi:10.24963/ijcai.2018/765`.

11 Stefan Borgwardt, İsmail İlkan Ceylan, and Thomas Lukasiewicz. Ontology-Mediated Query Answering over Log-Linear Probabilistic Data. In *Proceedings fo the Thirty-Third AAAI Conference on Artificial Intelligence*, volume 33, Palo Alto, CA, USA, 2019. AAAI Press. `doi:10.1609/aaai.v33i01.33012711`.

12 Jihad Boulos, Nilesh Dalvi, Bhushan Mandhani, Shobhit Mathur, Chris Ré, and Dan Suciu. MYSTIQ: A System for Finding more Answers by Using Probabilities. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data (SIGMOD '05)*, pages 891–893, New York, NY, USA, 2005. ACM. `doi:10.1145/1066157.1066277`.

13 Nicolas Bourbaki. *General Topology. Chapters 5–10.* Springer, Berlin and Heidelberg, Germany, 1st edition, 1989. Original French edition published by MASSON, Paris, 1974.

14 Nicolas Bourbaki. *General Topology. Chapters 1–4.* Springer, Berlin and Heidelberg, Germany, 1st edition, 1995. Original French edition published by MASSON, Paris, 1971. `doi:10.1007/978-3-642-61701-0`.

15 Roger Cavallo and Michael Pittarelli. The Theory of Probabilistic Databases. In *Proceedings of the 13th International Conference on Very Large Data Bases (VLDB '87)*, pages 71–81, San Francisco, CA, USA, 1987. Morgan Kaufmann.

16 İsmail İlkan Ceylan, Adnan Darwiche, and Guy Van den Broeck. Open-World Probabilistic Databases. In *Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning (KR '16)*, pages 339–348, Palo Alto, CA, USA, 2016. AAAI Press.

17 Reynold Cheng, Dmitri V. Kalashnikov, and Sunil Prabhakar. Evaluating Probabilistic Queries over Imprecise Data. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (SIGMOD '03)*, pages 551–562, New York, NY, USA, 2003. ACM. `doi:10.1145/872757.872823`.

18 Daryl John Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes, Volume I: Elementary Theory and Models.* Probability and its Applications. Springer, New York, NY, USA, 2nd edition, 2003. `doi:10.1007/b97277`.

19 Daryl John Daley and David Vere-Jones. *An Introduction to the Theory of Point Processes, Volume II: General Theory and Structure*. Probability and its Applications. Springer, New York, NY, USA, 2nd edition, 2008. doi:10.1007/978-0-387-49835-5.

20 Nilesh Dalvi, Christopher Ré, and Dan Suciu. Probabilistic Databases: Diamonds in the Dirt. *Communications of the ACM*, 52(7):86–94, 2009. doi:10.1145/1538788.1538810.

21 Umeshwar Dayal, Nathan Goodman, and Randy Howard Katz. An Extended Relational Algebra with Control over Duplicate Elimination. In *Proceedings of the 1st ACM SIGACT-SIGMOD Composium on Principles of Database Systems (PODS '82)*, pages 117–123, New York, NY, USA, 1982. ACM.

22 Luc De Raedt, Kristian Kersting, Sriraam Natarajan, and David Poole. *Statistical Relational Artificial Intelligence: Logic, Probability, and Computation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool, San Rafael, CA, USA, 2016.

23 Christoph Degen. *Finite Point Processes and Their Application to Target Tracking*. PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn, 2015.

24 Amol Deshpande, Carlos Guestrin, Samuel R. Madden, Joseph M. Hellerstein, and Wei Hong. Model-Driven Data Acquisition in Sensor Networks. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB '04)*, pages 588–599, St. Louis, 2004. Morgan Kaufmann. doi:10.1016/B978-012088469-8.50053-X.

25 Daniel Deutch, Christoph Koch, and Tova Milo. On Probabilistic Fixpoint and Markov Chain Query Languages. In *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS '10)*, pages 215–226, New York, NY, USA, 2010. ACM.

26 Debabrata Dey and Sumit Sarkar. A Probabilisitic Relational Model and Algebra. *ACM Transactions on Database Systems (TODS)*, 21(3), 1996. doi:10.1145/232753.232796.

27 Anton Faradjian, Johannes Gehrke, and Philippe Bonnett. GADT: A Probability Space ADT for Representing and Querying the Physical World. In *Proceedings of the 18th International Conference on Data Engineering (ICDE '02)*, pages 201—211. IEEE Computing Society, 2002. doi:10.1109/ICDE.2002.994710.

28 Robert Fink, Larisa Han, and Dan Olteanu. Aggregation in Probabilistic Databases via Knowledge Compilation. In *Proceedings of the 38th International Conference on Very Large Data Bases (VLDB '12)*, volume 5, pages 490–501. VLDB Endowment, 2012. doi:10.14778/2140436.2140445.

29 David H. Fremlin. *Measure Theory, Volume 4: Topological Measure Spaces*. Torres Fremlin, Colchester, UK, 2nd edition, 2013.

30 David H. Fremlin. *Measure Theory, Volume 2: Broad Foundations*. Torres Fremlin, Colchester, UK, 2nd printing edition, 2016.

31 Tal Friedman and Guy Van den Broeck. On Constrained Open-World Probabilistic Databases. In *The 1st Conference on Automated Knowledge Base Construction (AKBC)*, 2019.

32 Bert E. Fristedt and Lawrence F. Gray. *A Modern Approach to Probabilitiy Theory*. Probability and its Applications. Birkhäuser, Cambridge, MA, USA, 1st edition, 1997.

33 Norbert Fuhr. Probabilistic Datalog—A Logic for Powerful Retrieval Methods. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '95)*, pages 282–290, New York, NY, USA, 1995. ACM.

34 Norbert Fuhr and Thomas Rölleke. A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. *ACM Transactions on Information Systems (TOIS)*, 15(1):32–66, 1997. doi:10.1145/239041.239045.

35 Erol Gelenbe and George Hebrail. A Probability Model of Uncertainty in Data Bases. In *1986 IEEE Second International Conference on Data Engineering*, pages 328–333. IEEE, 1986 . doi:10.1109/ICDE.1986.7266237.

36 Todd J. Green. Models for Incomplete and Probabilistic Information. In Charu C. Aggarwal, editor, *Managing and Mining Uncertain Data*, volume 35 of *Advances in Database Systems*, chapter 2, pages 9–43. Springer, Boston, MA, USA, 2009. doi:10.1007/978-0-387-09690-2_2.

37 Martin Grohe and Peter Lindner. Infinite Probabilistic Databases, 2019. arXiv e-prints, arXiv:1904.06766 [cs.DB].

38 Martin Grohe and Peter Lindner. Probabilistic Databases with an Infinite Open-World Assumption. In *Proceedings of the 38th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS '19)*, pages 17–31, New York, NY, USA, 2019. ACM. Extended version available at arXiv: arXiv:1807.00607 [cs.DB]. doi:10.1145/3294052.3319681.

39 Stéphane Grumbach, Leonid Libkin, Tova Milo, and Limsoon Wong. Query Languages for Bags: Expressive Power and Complexity. *ACM SIGACT News*, 1996(2):30–44, 1996. doi:10.1145/235767.235770.

40 Stéphane Grumbach and Tova Milo. Towards Tractable Algebras for Bags. *Journal of Computer and System Sciences*, 52(3):570–588, 1996. doi:10.1006/jcss.1996.0042.

41 Ravi Jampani, Fei Xu, Mingxi Wu, Luis Perez, Chris Jermaine, and Peter J. Haas. MCDB: A Monte Carlo Approach to Managing Uncertain Data. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*, pages 687–700, New York, NY, USA, 2008. ACM Press. doi:10.1145/1376616.1376686.

42 Ravi Jampani, Fei Xu, Mingxi Wu, Luis Perez, Chris Jermaine, and Peter J. Haas. The Monte Carlo Database System: Stochastic Analysis Close to the Data. *ACM Transactions on Database Systems (TODS)*, 36(3):18:1–18:41, 2011. doi:10.1145/2000824.2000828.

43 Olav Kallenberg. *Foundations of Modern Probability*. Probability and its Applications. Springer, New York, NY, USA, 1st edition, 1997.

44 Oliver Kennedy and Christoph Koch. PIP: A Database System for Great and Small Expectations. In *Proceedings of the 26th International Conference on Data Engineering (ICDE '10)*, pages 157–168, Washington, DC, USA, 2010. IEEE.

45 Christoph Koch. On Query Algebras for Probabilistic Databases. *ACM SIGMOD Record*, 37(4):78–85, 2008.

46 Christoph Koch. MayBMS: A System for Managing Large Probabilistic Databases. In Charu C. Aggarwal, editor, *Managing and Mining Uncertain Data*, volume 35 of *Advances in Database Systems*, chapter 6, pages 149–184. Springer, Boston, MA, USA, 2009. doi:10.1007/978-0-387-09690-2_6.

47 Christoph Koch and Dan Olteanu. Conditioning Probabilistic Databases. In *Proceedings of the 34th International Conference on Very Large Data Bases (VLDB '08)*, volume 1, pages 313–325. VLDB Endowment, 2008. doi:10.14778/1453856.1453894.

48 Günter Last and Matthew Penrose. *Lectures on the Poisson Process*. Institute of Mathematical Statistics Textbook. Cambridge University Press, Cambridge, UK, 2017. doi:10.1017/9781316104477.

49 Odile Macchi. The Coincidence Approach to Stochastic Point Processes. *Advances in Applied Probability*, 7(1):83–122, 1975. doi:10.2307/1425855.

50 Ronald P. S. Mahler. *Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc., Norwood, MA, USA, 2007.

51 Brian Milch, Bhaskara Marthi, Stuart Russell, David Sontag, David L. Ong, and Andrey Kolobov. BLOG: Probabilistic Models with Unknown Objects. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI '05)*, St. Louis, MO, USA, 2005. Morgan Kaufmann.

52 Brian Christopher Milch. *Probabilistic Models with Unknown Objects*. PhD thesis, University of California, Berkeley, 2006.

53 José Enrique Moyal. The General Theory of Stochastic Population Processes. *Acta Mathematica*, 108:1–31, 1962. doi:10.1007/BF02545761.

54 Raghotham Murthy, Robert Ikeda, and Jennifer Widom. Making Aggregation Work in Uncertain and Probabilistic Databases. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 23(8):1261–1273, 2011. doi:10.1109/TKDE.2010.166.

55 Jian Pei, Bin Jiang, Xuemin Lin, and Yidong Yuan. Probabilistic Skylines on Uncertain Data. In *Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB '07)*, pages 15–26. VLDB Endowment, 2007.

56 Michael Pittarelli. An Algebra for Probabilistic Databases. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 6(2):293–303, 1994.

57 Raymond Reiter. On Closed World Data Bases. In Herve Gallaire and Jack Minker, editors, *Logic and Data Bases*, pages 55–76. Plenum Press, New York, NY, USA, 1st edition, 1978.

58 Matthew Richardson and Pedro Domingos. Markov Logic Networks. *Machine Learning*, 62(1–2):107—136, 2006. `doi:10.1007/s10994-006-5833-1`.

59 Robert Ross, V. S. Subrahmanian, and John Grant. Aggregate Operators in Probabilistic Databases. *Journal of the ACM (JACM)*, 52(1):54–101, 2005. `doi:10.1145/1044731.1044734`.

60 Sarvjeet Singh, Chris Mayfield, Rahul Shah, Sunil Prabhakar, Susanne Hambrusch, Jennifer Neville, and Reynold Cheng. Database Support for Probabilistic Attributes and Tuples. In *2008 IEEE 24th International Conference on Data Engineering (ICDE '08)*, pages 1053–1061, Washington, DC, USA, 2008. IEEE Computer Society. `doi:10.1109/ICDE.2008.4497514`.

61 Parag Singla and Pedro Domingos. Markov Logic in Infinite Domains. In *Proceedings of the Twenty-Third Conference on Uncertainty in Artificial Intelligence (UAI '07)*, pages 368–375, Arlington, VA, USA, 2007. AUAI Press.

62 Shashi Mohan Srivastava. *A Course on Borel Sets.* Graduate Texts in Mathematics. Springer, New York, NY, USA, 1st edition, 1998. `doi:10.1007/b98956`.

63 Dan Suciu, Dan Olteanu, Christopher Ré, and Christoph Koch. *Probabilistic Databases.* Synthesis Lectures on Data Management. Morgan & Claypool, San Rafael, CA, USA, 1st edition, 2011. `doi:10.2200/S00362ED1V01Y201105DTM016`.

64 Thanh T. L. Tran, Liping Peng, Yanlei Diao, Andrew McGregor, and Anna Liu. CLARO: Modeling and Processing Uncertain Data Streams. *The VLDB Journal*, 21(5):651–676, 2012. `doi:10.1007/s00778-011-0261-7`.

65 Guy Van den Broeck and Dan Suciu. Query Processing on Probabilistic Data: A Survey. *Foundations and Trends® in Databases*, 7(3–4):197–341, 2017. `doi:10.1561/1900000052`.

66 Brend Wanders and Maurice van Keulen. Revisiting the Formal Foundation of Probabilistic Databases. In *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT '15)*, Advances in Intelligent Systems Research, pages 289–296, Paris, France, 2015. Atlantis Press.

67 Yijie Wang, Xiaoyong Li, Xiaoling Li, and Yuan Wang. A Survey of Queries over Uncertain Data. *Knowledge and Information Systems*, 37(3):485–530, 2013. `doi:10.1007/s10115-013-0638-6`.

68 Jennifer Widom. Trio: A System for Data, Uncertainty, and Lineage. In Charu C. Aggarwal, editor, *Managing and Mining Uncertain Data*, volume 35 of *Advances in Database Systems*, chapter 5, pages 113–148. Springer, Boston, MA, USA, 2009. `doi:10.1007/978-0-387-09690-2_5`.

69 Eugene Wong. A Statistical Approach to Incomplete Information in Database Systems. *ACM Transactions on Database Systems (TODS)*, 7(3):470–488, 1982. `doi:10.1145/319732.319747`.

70 Yi Wu, Siddharth Srivastava, Nicholas Hay, Simon Du, and Stuart Russell. Discrete-Continuous Mixtures in Probabilistic Programming: Generalized Semantics and Inference Algorithms. In *Proceedings of the 35th International Conference on Machine Learning (ICML 2018)*, volume 80 of *Proceedings of Machine Learning Research*, pages 5343–5352. PMLR, 2018.

71 Esteban Zimányi. Query Evaluation in Probabilistic Relational Databases. *Theoretical Computer Science*, 171(1):179–219, 1997. `doi:10.1016/S0304-3975(96)00129-6`.