I N F S Y S
R E S E A R C H
R E P O R T

INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG WISSENSBASIERTE SYSTEME

A DATA MODEL AND ALGEBRA FOR PROBABILISTIC COMPLEX VALUES

Thomas Eiter   Thomas Lukasiewicz   Michael Walter

INFSYS RESEARCH REPORT 1843-00-04

AUGUST 2000

Institut für Informationssysteme
Abtg. Wissensbasierte Systeme
Technische Universität Wien
Favoritenstraße 9-11
A-1040 Wien, Austria
Tel: +43-1-58801-18405
Fax: +43-1-58801-18493
sek@kr.tuwien.ac.at
www.kr.tuwien.ac.at

TECHNISCHE UNIVERSITÄT WIEN

INFSYS RESEARCH REPORT

INFSYS RESEARCH REPORT 1843-00-04, AUGUST 2000

A DATA MODEL AND ALGEBRA FOR PROBABILISTIC COMPLEX VALUES

Thomas Eiter[1] Thomas Lukasiewicz[2] Michael Walter[3,4]

Abstract. We present a probabilistic data model for complex values. More precisely, we introduce probabilistic complex value relations, which combine the concept of probabilistic relations with the idea of complex values in a uniform framework. We elaborate a model-theoretic definition of probabilistic combination strategies, which has a rigorous foundation on probability theory. We then define an algebra for querying database instances, which comprises the operations of selection, projection, renaming, join, Cartesian product, union, intersection, and difference. We prove that our data model and algebra for probabilistic complex values generalizes the classical relational data model and algebra. Moreover, we show that under certain assumptions, all our algebraic operations are tractable. We finally show that most of the query equivalences of classical relational algebra carry over to our algebra on probabilistic complex value relations. Hence, query optimization techniques for classical relational algebra can easily be applied to optimize queries on probabilistic complex value relations.

[1]Institut und Ludwig Wittgenstein Labor für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Wien, Austria. E-mail: eiter@kr.tuwien.ac.at

[2]Institut und Ludwig Wittgenstein Labor für Informationssysteme, Technische Universität Wien, Favoritenstraße 9-11, A-1040 Wien, Austria. E-mail: lukasiew@kr.tuwien.ac.at

[3]IBM Germany, Überseering 24, D-22297 Hamburg, Germany. E-mail: MiWalter@de.ibm.com

[4]Main part of the work was carried out at the Institut für Informatik, Universität Gießen, Arndtstraße 2, D-35392 Gießen, Germany.

Acknowledgements: We are very grateful to V.S. Subrahmanian for fruitful discussions. Furthermore, we would like to thank the participants of the International Symposium on Foundations of Information and Knowledge Systems (FoIKS 2000), Burg (Spreewald), Germany, February 14–17, 2000, in particular Vladimir Sazonov and Paolo Ciaccia, for helpful comments and suggestions for improvements on a preliminary version of this paper.

This work has been partially supported by an NSF-DAAD grant, a DFG grant, and the Austrian Science Fund under Project N Z29-INF.

A preliminary version of this paper has been published under the title “Extension of Relational Algebra to Probabilistic Complex Values” in K.-D. Schewe and B. Thalheim, editors, *Proceedings International Symposium on Foundations of Information and Knowledge Systems (FoIKS 2000), Burg (Spreewald), Germany, February 14–17, 2000*, number 1762 in LNCS, pages 94–115. Springer, 2000.

Contents

1 Introduction

Databases are a central component of many business and information systems. During the past two decades, relational database systems have replaced earlier systems and have become a standard for data management. Various needs in practice, however, cannot be appropriately managed with current commercial database systems, which are largely based on the plain relational data model.

Since applications that involve uncertainty abound, an important such need is the integration of models of uncertainty into databases. Research on this issue can be divided into four categories [6]: (1) handling null values, (2) retrieval of incomplete data, (3) management of vague data (for example, "John is tall") in fuzzy set approaches, and (4) management of ambiguous or imprecise data (for example, "John's salary is between 50K and 70K") in probabilistic approaches. Another important issue is the management of structured objects. The relational model has been early generalized for storing structured objects [22], which was a step towards object-oriented database systems (see [1] for a background on complex values and a historic account).

In many applications, especially in the business domain, uncertainty stems from ambiguity rather than from vagueness. Several models for incorporating ambiguity into the relational model have thus been proposed so far (for example, [5, 3, 6, 17, 11]; see Section 7 for a discussion of these and further approaches). Informally, they have attached a probability to each tuple and/or to each value in a set of possible values of an imprecise attribute of a tuple. The operations of relational algebra are then generalized to combine these probabilities in a suitable way, adopting some underlying assumptions (like independence or disjointness of events).

In this paper, we extend this line of research to databases storing structured objects, and present a probabilistic data model for complex values. To our knowledge, not much work in this direction has been done so far. A pioneering paper is [10], which presented a probabilistic version of NF2 relations that builds on earlier work of the authors [11] under some limiting assumptions (see Section 7 for a discussion and comparison to our work).

Our data model model generalizes and refines a related model of annotated tuples [17] to complex values, which are built from atomic values applying constructors for tuple and set aggregation [1]. Informally, every complex value v is associated with a probability interval $[l, u]$ and an event e, forming a quadruple (v, l, u, e). The interval $[l, u]$ represents the likelihood that v belongs to a relation of the database, where e records information about how this value was derived.

The following is a simple example of a relation containing probabilistic complex values:

<table>
<tr><th colspan="2">v</th><th>l</th><th>u</th><th>e</th></tr>
<tr><td>patient</td><td>diseases</td><td rowspan="2">0.7</td><td rowspan="2">0.9</td><td rowspan="2">$e_1 \vee e_2$</td></tr>
<tr><td>John</td><td>$\{$lung cancer, tuberculosis$\}$</td></tr>
<tr><td>patient</td><td>diseases</td><td rowspan="2">0.5</td><td rowspan="2">0.7</td><td rowspan="2">e_3</td></tr>
<tr><td>Jack</td><td>$\{$leprosy$\}$</td></tr>
</table>

Intuitively, this relation holds data about patients and their diseases. The probability that John suffers from lung cancer and tuberculosis is between 0.7 and 0.9, while the probability that Jack suffers from leprosy is between 0.5 and 0.7. The associated events $e_1 \vee e_2$ and e_3, respectively, show us that the former tuple must have been derived by combining tuples (leading to a disjunction of events), while the latter tuple may stem from a single tuple in the database. Here, we have interpreted the probabilistic information as subjective belief about the diseases that persons suffer from. However, in general this information might be interpreted in different ways; we defer a discussion of this issue to the end of Section 3 (see Example 3.3).

We observe that in the above model, probabilities are assigned to a complex value as a whole, and no impreciseness in attributes, given by sets of values with probabilities attached, can be explicitly expressed in the language. It has been shown in [17] that each tuple with imprecise attribute values can be represented by an annotated tuple, and that such a representation can be efficiently computed. Applying similar techniques, complex values with imprecise attributes may be represented by probabilistic complex values as above. To keep our model simple, we do not consider imprecise attributes here.

On top of our model, we define a relational algebra. Following the approach in [17], it uses generic functions $\otimes$, $\oplus$, $\ominus$ for computing the probability range of the conjunction, disjunction, and difference of two events e_1 and e_2 from the probability ranges $[l_1, u_1]$ and $[l_2, u_2]$ of e_1 and e_2, respectively. Instances of these functions are selected according to the relationship between e_1 and e_2. For example, if e_1 and e_2 are independent, then the probability range of $e_1 \wedge e_2$ is given by $[l_1 \cdot l_2, u_1 \cdot u_2]$. However, if nothing is known about how e_1 and e_2 relate, then it is given by $[\max(0, l_1 + l_2 - 1), \min(u_1, u_2)]$. Such generic combination functions allow us to remove the explicit or implicit assumptions about joint occurrence of tuples and/or attribute values in relations that are present in other approaches (see [5, 3, 6, 11]). As we shall discuss below, our model of combination functions advances and refines the one in [17].

We remark that, similar to [17], our model uses interval probabilities rather than point probabilities. The reasons are twofold:

- Probability intervals are a generalization of point probabilities, and thus they provide a greater flexibility. In particular, they allow one for expressing imprecise probabilistic knowledge. Hence, probability intervals are better suited especially to represent statistical knowledge and subjective degrees of belief. More precisely, statistical findings are almost always associated with imprecise intervals, rather than with exact point values. Moreover, rather than giving a complete and precise picture about the world, our subjective knowledge is often incomplete and imprecise (see especially [16] for a survey on this issue). For example, the information that the probability of a certain fact being true is at least 0.5 (for instance, that patient Jill suffers from insomnia) can be easily expressed through the probability interval [0.5,1], and similarly for the case where the probability is at most a certain value. Moreover, we can explicitly express that the probability of a certain fact is unknown by using the probability interval [0,1]. This is in particular useful, if the data model adopts the closed world assumption, that is, if we assume that all facts that are not mentioned in the database are false (that is, have probability 0).

- Probability intervals are needed for technical reasons, since our probabilistic knowledge, even when using only point probabilities, generally specifies *a set* of probability distributions, rather than *a unique single* distribution. For example, given the point probabilities p_1 and p_2 of the events e_1 and e_2, respectively, and assuming that nothing is known about the relationship between e_1 and e_2, we can only conclude that the probability of $e_1 \wedge e_2$ is contained in the interval $[\max(0, p_1 + p_2 - 1), \min(p_1, p_2)]$.

The main contributions of this work can be briefly summarized as follows:

- To elucidate their relation and underlying connection to probability theory, we provide a model-theoretic definition of probabilistic combination strategies as in [17], which assumes a probabilistic semantics in which probabilities are defined over a set of possible worlds. Moreover, differently from [17], we propose probabilistic difference strategies that are not necessarily derived from conjunction strategies and negation. As we show, these refinements lead to more precise results in certain cases (see Example 7.3).

- We present a data model that is based on probabilistic complex value relations. This model generalizes previous data models for probabilistic relational databases [17] and complex value databases in the literature [1].

- We define an algebra for querying database instances. We prove that our algebra generalizes the familiar relational algebra for relational databases. Moreover, we show that under certain assumptions, all our algebraic operations are tractable.

- We present a host of equivalence results for query expressions in our algebra. Since our model generalizes the one in [17], our results also apply to [17] as a special case. Note that the results on query equivalences in [17] are limited to the relationship between compaction and the standard relational algebra operators.

- Using the result that many query equivalences of classical relational algebra carry over to our algebra on probabilistic complex value relations, we show that query optimization techniques for classical relational algebra can be easily generalized to optimize queries on probabilistic complex value relations.

The rest of this paper is organized as follows. After giving some preliminaries, we develop in Section 2 our model for representing and processing probabilistic knowledge, which comprises probabilistic dependence between complex values, probabilistic information about complex values, and probabilistic combination strategies. In Section 3 we define our data model, and in Section 4 we present the algebra on it. Section 5 investigates query equivalences in the algebra, which are applied in Section 6 for query optimization. In Section 7 we review related work and in particular compare our model to ProbView [17]. In the final Section 8, we discuss further issues and conclude the paper. To increase readability, proofs of technical results have been moved to the appendix.

2 Probabilistic Background

In this section, we describe the probabilistic background of our approach to probabilistic complex value databases. We assume a semantics in which probabilities are defined over a set of possible worlds (see especially [4, 12, 23, 14]), where we adopt some technical notions from [19, 20].

A major goal of this section is to give a model-theoretic definition of probabilistic conjunction, disjunction, and difference strategies, which have been introduced by an axiomatic characterization in [17]. Given the probability ranges of two events e_1 and e_2, these strategies compute the probability range of the events $e_1 \vee e_2$, $e_1 \wedge e_2$, and $e_1 \wedge \neg e_2$, respectively. We allow a variety of different probabilistic conjunction, disjunction, and difference strategies to take into account the different dependencies between e_1 and e_2. We benefit from this definition by a better understanding of the constraints that are imposed in the axiomatic approach, and unhide also some implicit assumptions in [17].

2.1 Probabilistic Knowledge

We assume a set of *basic events* $\mathcal{B} = \{b_1, b_2, \ldots, b_n\}$ with $n \geq 1$. The set of *events* is the closure of $\mathcal{B}$ under the Boolean connectives $\neg$ and $\wedge$. As usual, we use $(e_1 \vee e_2)$ to abbreviate $\neg(\neg e_1 \wedge \neg e_2)$. We use $\bot$ and $\top$ to abbreviate the *false event* $(b_1 \wedge \neg b_1)$ and the *true event* $\neg(b_1 \wedge \neg b_1)$, respectively. A *probabilistic pair* is an expression of the form $(e, [l, u])$, where l and u are real numbers from $[0, 1]$ and e is an event. We use $\bot$ and $\top$ to abbreviate $(\bot, [0, 0])$ and $(\top, [1, 1])$, respectively.

Informally, each tuple t in a probabilistic complex value database will be associated with a probabilistic pair $(e, [l, u])$. Intuitively, this means that t is identified with e and that the probability of t lies in the interval range $[l, u]$. More precisely, each tuple in a base relation will be assigned a probabilistic pair $(b, [l, u])$ with a basic event b. Each derived tuple t will be assigned a probabilistic pair $(e, [l, u])$ with a general event e, which encodes some information on how t is computed from tuples in the base relations.

A *dependence formula* is an expression of form $e_1 \rightarrow e_2$ (called *implication*), $NC(e_1, e_2)$ (*negative correlation*), $PC(e_1, e_2)$ (*positive correlation*), or $Ind(e_1, e_2)$ (*independence*) where e_1 and e_2 are events. A *dependence information* on two events e_1 and e_2 is any subset $KB \subseteq KB^\star(e_1, e_2)$ where

$$KB^\star(e_1, e_2) = \{Ind(e_1, e_2),\ NC(e_1, e_2),\ PC(e_1, e_2),\ e_1 \rightarrow e_2,\ e_2 \rightarrow e_1,\ e_1 \wedge e_2 \rightarrow \bot,\ \top \rightarrow e_1 \vee e_2\}.$$

An *event formula* (or shortly *formula*) is a probabilistic pair or a dependence formula.

Informally, to express that the probabilities of two events e_1 and e_2 lie in the intervals $[l_1, u_1]$ and $[l_2, u_2]$, respectively, we will use the two probabilistic pairs $(e_1, [l_1, u_1])$ and $(e_2, [l_2, u_2])$, respectively. The relationship between e_1 and e_2 will then be encoded by some dependence information $KB \subseteq KB^\star(e_1, e_2)$.

A *classical interpretation* I is a standard truth assignment $I : \mathcal{B} \rightarrow \{$ true, false$\}$, which is extended to all events as usual (that is, $(e_1 \wedge e_2)$ is true in I iff e_1 and e_2 are true in I, and $\neg e$ is true in I iff e is not true in I). We write $I \models e$ iff e is true in I, and denote by $\mathcal{I}_\mathcal{B}$ the set of all classical interpretations on $\mathcal{B}$.

A *probabilistic interpretation* Pr is a mapping $Pr\colon \mathcal{I}_\mathcal{B} \rightarrow [0, 1]$ such that all $Pr(I)$ with $I \in \mathcal{I}_\mathcal{B}$ sum up to 1. It is extended to all events e as follows:

$$Pr(e) \ = \sum_{I \in \mathcal{I}_\mathcal{B},\, I \models e} Pr(I)\,.$$

It is important to point out that probabilistic interpretations are defined on the set of all classical interpretations and not on the set of all basic events. That is, we do *not* assume that the basic events are pairwise mutually exclusive. Moreover, we do *not* assume that the basic events are pairwise independent.

The *truth* of a formula F in a probabilistic interpretation Pr, denoted $Pr \models F$, is defined as follows:

$$\begin{array}{lll}
Pr \models (e, [l, u]) & \text{iff} & Pr(e) \in [l, u]\,, \\
Pr \models e_1 \rightarrow e_2 & \text{iff} & Pr(e_1 \wedge e_2) = Pr(e_1)\,, \\
Pr \models NC(e_1, e_2) & \text{iff} & Pr(e_1 \wedge \neg e_2) = \min(Pr(e_1), Pr(\neg e_2))\,, \\
Pr \models PC(e_1, e_2) & \text{iff} & Pr(e_1 \wedge e_2) = \min(Pr(e_1), Pr(e_2))\,, \\
Pr \models Ind(e_1, e_2) & \text{iff} & Pr(e_1 \wedge e_2) = Pr(e_1) \cdot Pr(e_2)\,.
\end{array}$$

A probabilistic interpretation Pr is a *model* of a formula F, or Pr *satisfies* F, iff $Pr \models F$. It is a *model* of a set of formulas $\mathcal{F}$ (or Pr *satisfies* $\mathcal{F}$, denoted $Pr \models \mathcal{F}$) iff Pr is a model of all $F \in \mathcal{F}$. The set $\mathcal{F}$ is *satisfiable* iff a model of $\mathcal{F}$ exists. A formula F is a *logical consequence* of $\mathcal{F}$, denoted $\mathcal{F} \models F$, iff each model of $\mathcal{F}$ is also a model of F. Two sets of formulas $\mathcal{F}$ and $\mathcal{F}'$ are *logically equivalent* iff they have the same models.

A probabilistic pair $(e, [l, u])$ is a *tight logical consequence* of a set of formulas $\mathcal{F}$, denoted $\mathcal{F} \models_{tight} (e, [l, u])$, iff l (resp., u) is the infimum (resp., supremum) of $Pr(e)$ subject to all models Pr of $\mathcal{F}$. Note that we canonically define $\mathcal{F} \models_{tight} (e, [1, 0])$ when $\mathcal{F}$ is unsatisfiable.

Among the many possible dependence informations $KB \subseteq KB^\star(a, b)$ on distinct basic events a and b, the following deserve special attention: $\emptyset$ (called *ignorance*), $\{Ind(a, b)\}$ (*independence*), $\{PC(a, b)\}$ (*positive correlation*), $\{b \rightarrow a\}$ (*left implication*), $\{a \rightarrow b\}$ (*right implication*), $\{b \rightarrow a,\ a \rightarrow b\}$ (*equivalence*),

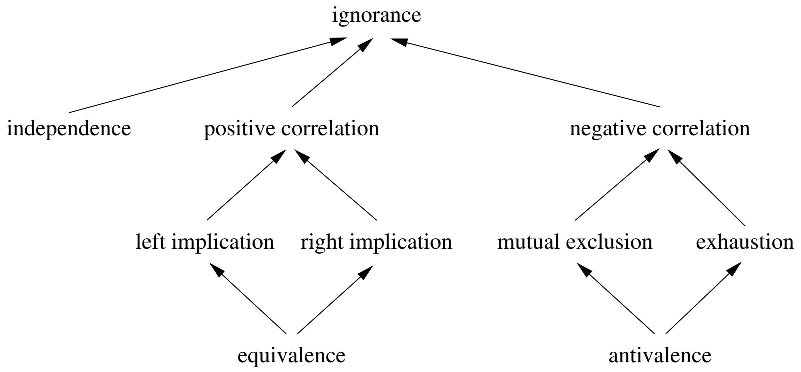

Figure 1: Logical implication of dependence information

$\{NC(a,b)\}$ (*negative correlation*), $\{a \wedge b \rightarrow \bot\}$ (*mutual exclusion*), $\{\top \rightarrow a \vee b\}$ (*exhaustion*), and $\{a \wedge b \rightarrow \bot,\ \top \rightarrow a \vee b\}$ (*antivalence*). Figure 1 shows their order diagram according to logical implication. The following proposition states some relationships between the dependence informations in Figure 1.

Proposition 2.1 Let a and b be two distinct basic events. Then, for every probabilistic interpretation Pr, the following holds:

$$Pr \models PC(a,b) \quad \text{iff} \quad Pr \models a \rightarrow b \ \text{ or } \ Pr \models b \rightarrow a\,. \tag{2.1}$$

$$Pr \models NC(a,b) \quad \text{iff} \quad Pr \models a \wedge b \rightarrow \bot \ \text{ or } \ Pr \models \top \rightarrow a \vee b\,. \tag{2.2}$$

$$\text{If } Pr \models F \text{ for some } F \in \{a \rightarrow b,\ b \rightarrow a,\ PC(a,b)\}, \text{ then} \tag{2.3}$$
$$Pr \models NC(a,b) \ \text{ iff } \ Pr \models Ind(a,b)\,.$$

$$\text{If } Pr \models F \text{ for some } F \in \{a \wedge b \rightarrow \bot,\ \top \rightarrow a \vee b,\ NC(a,b)\}, \text{ then} \tag{2.4}$$
$$Pr \models PC(a,b) \ \text{ iff } \ Pr \models Ind(a,b)\,.$$

Exploiting this proposition, we show that every $KB \subseteq KB^\star(a,b)$ except independence is logically equivalent to a logical disjunction of dependence informations KB_i which may include only left and right implication, exhaustion, and antivalence. Note that the KB_i's can be efficiently constructed from KB.

Theorem 2.2 Let a and b be two distinct basic events. For every dependence information $KB \subseteq KB^\star(a,b)$ with $KB \neq \{Ind(a,b)\}$, there are sets $KB_1,\ldots,KB_n \subseteq \{a \rightarrow b,\ b \rightarrow a,\ a \wedge b \rightarrow \bot,\ \top \rightarrow a \vee b\}$, where $1 \leq n \leq 4$, such that for every probabilistic interpretation Pr it holds that

$$Pr \models KB \ \text{ iff } \ Pr \models KB_1 \text{ or } \cdots \text{ or } Pr \models KB_n\,.$$

From the proof of this result, we obtain the following result about meaningful dependence information.

Lemma 2.3 Let a and b be two distinct basic events, and let $KB \subseteq KB^\star(a,b)$. Then, KB is satisfiable iff $\{a \rightarrow b,\ b \rightarrow a,\ a \wedge b \rightarrow \bot,\ \top \rightarrow a \vee b\} \not\subseteq KB$.

2.2 Satisfiability of Probabilistic Pairs

We now concentrate on the satisfiability of probabilistic pairs under dependence information.

The following proposition provides some simple conditions which characterize satisfiability of two probabilistic pairs over distinct basic events under several different cases of dependence information. In particular, it shows that there are combinations of probabilistic pairs $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])\}$ with dependence information $KB \subseteq KB^\star(e_1, e_2)$ that are unsatisfiable.

Proposition 2.4 Let a and b be two distinct basic events. Let $p_1 = (a, [l_1, u_1])$ and $p_2 = (b, [l_2, u_2])$ be two probabilistic pairs with $l_1 \leq u_1$ and $l_2 \leq u_2$.

(a) Suppose $KB \in \{\emptyset, \{Ind(a,b)\}, \{PC(a,b)\}, \{NC(a,b)\}\}$. Then, $KB \cup \{p_1, p_2\}$ is satisfiable.

(b) Suppose $KB = \{a \rightarrow b\}$. Then, $KB \cup \{p_1, p_2\}$ is satisfiable iff $l_1 \leq u_2$.

(c) Suppose $KB = \{b \rightarrow a\}$. Then, $KB \cup \{p_1, p_2\}$ is satisfiable iff $l_2 \leq u_1$.

(d) Suppose $KB = \{a \rightarrow b,\, b \rightarrow a\}$. Then, $KB \cup \{p_1, p_2\}$ is satisfiable iff $l_1 \leq u_2$ and $l_2 \leq u_1$.

(e) Suppose $KB = \{a \wedge b \rightarrow \bot\}$. Then, $KB \cup \{p_1, p_2\}$ is satisfiable iff $l_1 + l_2 \leq 1$.

(f) Suppose $KB = \{\top \rightarrow a \vee b\}$. Then, $KB \cup \{p_1, p_2\}$ is satisfiable iff $u_1 + u_2 \geq 1$.

(g) Suppose $KB = \{a \wedge b \rightarrow \bot, \top \rightarrow a \vee b\}$. Then, $KB \cup \{p_1, p_2\}$ is satisfiable iff $l_1 + l_2 \leq 1$ and $u_1 + u_2 \geq 1$.

We give an illustrative example.

Example 2.5 Let $p_1 = (a, [.5, 1])$, $p_2 = (b, [.6, 1])$, and $KB = \{a \wedge b \rightarrow \bot\}$. Then, $KB \cup \{p_1, p_2\}$ is unsatisfiable, since $0.5 + 0.6 > 1$. Intuitively, the lower bound 0.6 for the probability of b implies the upper bound $0.4 < 0.5$ for the probability of a.

The next example shows that unsatisfiability may also originate from the dependence information that is implicitly encoded in the logical structure of the events in the two probabilistic pairs.

Example 2.6 Let $p_1 = (e, [.5, 1])$, $p_2 = (\neg e, [.6, 1])$, and $KB = \emptyset$. Then, $KB \cup \{p_1, p_2\}$ is unsatisfiable, since $KB \models e \wedge \neg e \rightarrow \bot$ and $0.5 + 0.6 > 1$. Intuitively, as in Example 2.5, the lower bound 0.6 for the probability of $\neg e$ implies the upper bound $0.4 < 0.5$ for the probability of e.

2.3 Probabilistic Combination Strategies

Based on dependence information, we define now static and dynamic probabilistic combination strategies, which compute the probability of certain Boolean combination of events.

Definition 2.7 Let $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])$ be two probabilistic pairs and let KB be a dependence information on e_1 and e_2 such that $KB \cup \{p_1, p_2\}$ is satisfiable. The *probabilistic conjunction* (resp., *disjunction*, *difference*) of p_1 and p_2 under KB, denoted $p_1 \otimes_{KB} p_2$ (resp., $p_1 \oplus_{KB} p_2$, $p_1 \ominus_{KB} p_2$), is defined as the probabilistic pair $(e_1 \wedge e_2, [l, u])$ (resp., $(e_1 \vee e_2, [l, u])$, $(e_1 \wedge \neg e_2, [l, u])$), where $[l, u]$ is such that $(e_1 \wedge e_2, [l, u])$ (resp., $(e_1 \vee e_2, [l, u])$, $(e_1 \wedge \neg e_2, [l, u])$) is a tight logical consequence of $KB \cup \{p_1, p_2\}$.

Informally, to compute the probability range $[l, u]$ of $e_1 \wedge e_2$ from the probability ranges $[l_1, u_1]$ and $[l_2, u_2]$ of e_1 and e_2, respectively, we first collect the available dependence information KB on e_1 and e_2. We then check whether $KB \cup \{(e_1, [l_1, u_1]), (e_2, [l_2, u_2])\}$ is satisfiable. If this is the case, we compute the sharpest interval $[l, u]$ for $e_1 \wedge e_2$ which is possible given KB. This interval is given by the (unique) tight logical consequence $(e_1 \wedge e_2, [l, u])$ of $KB \cup \{(e_1, [l_1, u_1]), (e_2, [l_2, u_2])\}$. The probability range of $e_1 \vee e_2$ and $e_1 \wedge \neg e_2$ is computed analogously.

Note that, by the structure of $KB \cup \{p_1, p_2\}$, both l and u are rational numbers if l_1, l_2, u_1, and u_2 are, and they are the minimum and the maximum, respectively, of $Pr(e_1 \wedge e_2)$ subject to all models Pr of $KB \cup \{p_1, p_2\}$.

We now introduce probabilistic conjunction, disjunction, and difference *strategies*. More precisely, we introduce the concepts of *static* and *dynamic* such strategies, which differ in the way dependence information is used. This is motivated by the algebra introduced in Section 4, and briefly explained as follows.

In query expressions of our algebra, each occurrence of a probabilistic combination operator stands for the probabilistic combination of several pairs of probabilistic pairs. Moreover, each such occurrence will be parameterized with a dependence information $KB \subseteq KB^\star(a, b)$ on two new distinct basic events a and b. For example, $r_1 \cup^{\oplus_{in}} r_2$ will denote the union of two relations r_1 and r_2 under $KB = \{Ind(a, b)\}$. This dependence information KB can now be used in the following two ways:

STATIC DEPENDENCE: Given two probabilistic pairs $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_1, u_2])$, we compute $q_1 \odot q_2$, where $\odot \in \{\otimes_{KB}, \oplus_{KB}, \ominus_{KB}\}$, $q_1 = (a, [l_1, u_1])$, and $q_2 = (b, [l_1, u_2])$. That is, we consider only the dependence information in KB and simply ignore the concrete events e_1 and e_2.

DYNAMIC DEPENDENCE: Given two probabilistic pairs $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_1, u_2])$, we compute $p_1 \odot p_2$, where $\odot \in \{\otimes_{KB'}, \oplus_{KB'}, \ominus_{KB'}\}$ and KB' is obtained from KB by replacing a (resp., b) by e_1 (resp., e_2). That is, we consider the explicit dependence information in KB as well as the implicit dependence information that is encoded in the structure of the concrete events e_1 and e_2.

Inspection of the semantical and computational properties of these notions shows the following.

- STATIC DEPENDENCE has nice computational properties. First, the dependence information KB is in itself complete (that is, we do not have to compute any other relationships that are implicitly encoded in the structure of some concrete events e_1 and e_2). As we shall show in Section 4.5, under STATIC DEPENDENCE, $p_1 \odot p_2$ can be computed efficiently in constant time (that is, in constantly many steps). Second, some specific KB often implies nice algebraic properties of $\otimes_{KB}$, $\oplus_{KB}$, and $\ominus_{KB}$ that are known in advance, and can thus be exploited for query optimization (see Section 6).

- DYNAMIC DEPENDENCE, in contrast, has nice properties from the semantic point of view. It exploits all the (locally) available dependence information, and yields a more precise result than STATIC DEPENDENCE. However, it is easy to see that under DYNAMIC DEPENDENCE, computing $p_1 \odot p_2$ is intractable, which is inherited from the intractability of evaluating Boolean formulas.

We now define static and dynamic probabilistic combination strategies.

Definition 2.8 Let $KB \subseteq KB^\star(a, b)$ with two distinct basic events a and b. The *static probabilistic conjunction* (resp., *disjunction*, *difference*) *strategy* for KB is the unique function $\otimes$ (resp., $\oplus$, $\ominus$) that maps any $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])$ such that $KB \cup \{q_1, q_2\}$ is satisfiable to $q_1 \otimes_{KB} q_2$ (resp., $q_1 \oplus_{KB} q_2$, $q_1 \ominus_{KB} q_2$), where $q_1 = (a, [l_1, u_1])$ and $q_2 = (b, [l_2, u_2])$. The *dynamic probabilistic conjunction* (resp., *disjunction*, *difference*) *strategy* for KB is the unique function $\otimes$ (resp., $\oplus$, $\ominus$) that associates any two probabilistic pairs $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])$ such that $KB' \cup \{p_1, p_2\}$ is satisfiable with $p_1 \otimes_{KB'} p_2$ (resp., $p_1 \oplus_{KB'} p_2$, $p_1 \ominus_{KB'} p_2$), where KB' is obtained from KB by replacing a (resp., b) by e_1 (resp., e_2).

The static probabilistic combination strategies for the dependence informations in Figure 1 are shown in Table 1. The following proposition states that it is correct.

Table 1: Examples of static probabilistic combination strategies

ignorance $KB = \emptyset$	$p_1 \otimes_{ig} p_2 \; = \; (e_1 \wedge e_2, [\max(0, l_1 + l_2 - 1), \min(u_1, u_2)])$ $p_1 \oplus_{ig} p_2 \; = \; (e_1 \vee e_2, [\max(l_1, l_2), \min(1, u_1 + u_2)])$ $p_1 \ominus_{ig} p_2 \; = \; (e_1 \wedge \neg e_2, [\max(0, l_1 - u_2), \min(u_1, 1 - l_2)])$
independence $KB = \{Ind(a, b)\}$	$p_1 \otimes_{in} p_2 \; = \; (e_1 \wedge e_2, [l_1 \cdot l_2, u_1 \cdot u_2])$ $p_1 \oplus_{in} p_2 \; = \; (e_1 \vee e_2, [l_1 + l_2 - l_1 \cdot l_2, u_1 + u_2 - u_1 \cdot u_2])$ $p_1 \ominus_{in} p_2 \; = \; (e_1 \wedge \neg e_2, [l_1 \cdot (1 - u_2), u_1 \cdot (1 - l_2)])$
positive correlation $KB = \{PC(a, b)\}$	$p_1 \otimes_{pc} p_2 \; = \; (e_1 \wedge e_2, [\min(l_1, l_2), \min(u_1, u_2)])$ $p_1 \oplus_{pc} p_2 \; = \; (e_1 \vee e_2, [\max(l_1, l_2), \max(u_1, u_2)])$ $p_1 \ominus_{pc} p_2 \; = \; (e_1 \wedge \neg e_2, [\max(0, l_1 - u_2), \max(0, u_1 - l_2)])$
left implication $KB = \{b \to a\}$	$p_1 \otimes_{li} p_2 \; = \; (e_1 \wedge e_2, [l_2, \min(u_1, u_2)])$ $p_1 \oplus_{li} p_2 \; = \; (e_1 \vee e_2, [\max(l_1, l_2), u_1])$ $p_1 \ominus_{li} p_2 \; = \; (e_1 \wedge \neg e_2, [\max(0, l_1 - u_2), u_1 - l_2])$
right implication $KB = \{a \to b\}$	$p_1 \otimes_{ri} p_2 \; = \; (e_1 \wedge e_2, [l_1, \min(u_1, u_2)])$ $p_1 \oplus_{ri} p_2 \; = \; (e_1 \vee e_2, [\max(l_1, l_2), u_2])$ $p_1 \ominus_{ri} p_2 \; = \; (e_1 \wedge \neg e_2, [0, 0])$
equivalence $KB = \{a \to b,$ $b \to a\}$	$p_1 \otimes_{eq} p_2 \; = \; (e_1 \wedge e_2, [\max(l_1, l_2), \min(u_1, u_2)])$ $p_1 \oplus_{eq} p_2 \; = \; (e_1 \vee e_2, [\max(l_1, l_2), \min(u_1, u_2)])$ $p_1 \ominus_{eq} p_2 \; = \; (e_1 \wedge \neg e_2, [0, 0])$
negative correlation $KB = \{NC(a, b)\}$	$p_1 \otimes_{nc} p_2 \; = \; (e_1 \wedge e_2, [\max(0, l_1 + l_2 - 1), \max(0, u_1 + u_2 - 1)])$ $p_1 \oplus_{nc} p_2 \; = \; (e_1 \vee e_2, [\min(1, l_1 + l_2), \min(1, u_1 + u_2)])$ $p_1 \ominus_{nc} p_2 \; = \; (e_1 \wedge \neg e_2, [\min(l_1, 1 - u_2), \min(u_1, 1 - l_2)])$
mutual exclusion $KB = \{a \wedge b \to \bot\}$	$p_1 \otimes_{me} p_2 \; = \; (e_1 \wedge e_2, [0, 0])$ $p_1 \oplus_{me} p_2 \; = \; (e_1 \vee e_2, [l_1 + l_2, \min(1, u_1 + u_2)])$ $p_1 \ominus_{me} p_2 \; = \; (e_1 \wedge \neg e_2, [l_1, \min(u_1, 1 - l_2)])$
exhaustion $KB = \{\top \to a \vee b\}$	$p_1 \otimes_{ex} p_2 \; = \; (e_1 \wedge e_2, [\max(0, l_1 + l_2 - 1), u_1 + u_2 - 1])$ $p_1 \oplus_{ex} p_2 \; = \; (e_1 \vee e_2, [1, 1])$ $p_1 \ominus_{ex} p_2 \; = \; (e_1 \wedge \neg e_2, [1 - u_2, \min(u_1, 1 - l_2)])$
antivalence $KB = \{a \wedge b \to \bot,$ $\top \to a \vee b\}$	$p_1 \otimes_{an} p_2 \; = \; (e_1 \wedge e_2, [0, 0])$ $p_1 \oplus_{an} p_2 \; = \; (e_1 \vee e_2, [1, 1])$ $p_1 \ominus_{an} p_2 \; = \; (e_1 \wedge \neg e_2, [\max(l_1, 1 - u_2), \min(u_1, 1 - l_2)])$

Proposition 2.9 Let $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])$ be two probabilistic pairs. The static probabilistic combination strategies for ignorance, independence, positive correlation, left implication, right implication, equivalence, negative correlation, mutual exclusion, exhaustion, and antivalence are given as in Table 1.

We remark that the static probabilistic conjunction and disjunction strategies for ignorance, independence, positive correlation, and mutual exclusion are already known from Baldwin [2]. Recently, the probabilistic conjunction and disjunction strategies for ignorance, independence, positive correlation, negative correlation, and mutual exclusion have especially been discussed in [18] and [17]. To our knowledge, the strategies for the remaining dependence informations in Table 1 have not been considered so far.

2.4 Properties of Probabilistic Combination Strategies

We next focus on the properties of probabilistic combination strategies, and start with some useful notations.

For probabilistic pairs $p = (e, [l, u])$, we write $\iota(p)$ to denote $[l, u]$. For intervals $[r_1, s_1], [r_2, s_2] \subseteq [0, 1]$, we write $[r_1, s_1] \leq [r_2, s_2]$ to denote $r_1 \leq r_2$ and $s_1 \leq s_2$. For probabilistic pairs p_1 and p_2, we write $p_1 \subseteq p_2$, $p_1 \equiv p_2$, $p_1 \leq p_2$, and $p_1 \geq p_2$ to denote $\iota(p_1) \subseteq \iota(p_2)$, $\iota(p_1) = \iota(p_2)$, $\iota(p_1) \leq \iota(p_2)$, and $\iota(p_1) \geq \iota(p_2)$, respectively.

A (static or dynamic) probabilistic conjunction or disjunction strategy $\odot$ is *commutative* iff $p_1 \odot p_2 \equiv p_2 \odot p_1$ for all probabilistic pairs p_1 and p_2. It is *associative* iff $(p_1 \odot p_2) \odot p_3 \equiv p_1 \odot (p_2 \odot p_3)$ for all probabilistic pairs p_1, p_2, and p_3. A probabilistic conjunction strategy $\otimes$ is *distributive* over a probabilistic disjunction strategy $\oplus$ iff $p_1 \otimes (p_2 \oplus p_3) = (p_1 \otimes p_2) \oplus (p_1 \otimes p_3)$ for all p_1, p_2, and p_3.

Following [17], we say that $\oplus$ has the *identity* property iff $p \oplus \perp = p$ for all probabilistic pairs p. We say that $\otimes$ (resp., $\oplus$) has the *bottomline* property iff $p_1 \otimes p_2 \leq p_1 \otimes_{pc} p_2$ (resp., $p_1 \oplus p_2 \geq p_1 \oplus_{pc} p_2$) for all probabilistic pairs p_1 and p_2.

For associative static or dynamic probabilistic disjunction strategies $\oplus$ and probabilistic pairs $p_1, \ldots, p_k$ with $k \geq 1$, we write $\bigoplus_{i \in [1:k]} p_i$ to denote $p_1 \oplus \cdots \oplus p_k$.

The following proposition identifies some static probabilistic conjunction and disjunction strategies in Table 1 that are commutative, associative, and distributive.

Proposition 2.10 The static probabilistic conjunction and disjunction strategies for ignorance, independence, positive correlation, equivalence, negative correlation, mutual exclusion, exhaustion, and antivalence are all commutative and associative. Moreover, for positive correlation, the static conjunction strategy is distributive over the static disjunction strategy.

Commutativity of these strategies generalizes to the dynamic case:

Proposition 2.11 The dynamic probabilistic conjunction and disjunction strategies for ignorance, independence, positive correlation, equivalence, negative correlation, mutual exclusion, exhaustion, and antivalence are all commutative.

We close this section with describing some further properties of static probabilistic combination strategies.

Proposition 2.12 Let $KB \subseteq KB^{\star}(a, b)$ with two distinct basic events a and b. Let $\otimes$, $\oplus$, and $\ominus$ be the static probabilistic combination strategies for KB. Then, the conditions shown in Table 2 hold for all probabilistic pairs $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])$ such that $KB \cup \{(a, [l_1, u_1]), (b, [l_2, u_2])\}$ is satisfiable.

That is, ignorance is always an envelope for a sound approximation of the combination of p_1 and p_2. Moreover, probabilistic pairs with probability 0 (i.e., probability interval [0,0]) and 1 (i.e., [1,1]) serve under

Table 2: Properties of static probabilistic combination strategies

IGNORANCE	$p_1 \otimes p_2 \ \subseteq\ p_1 \otimes_{ig} p_2$ $p_1 \oplus p_2 \ \subseteq\ p_1 \oplus_{ig} p_2$ $p_1 \ominus p_2 \ \subseteq\ p_1 \ominus_{ig} p_2$
WEAK IDENTITY	$p_1 \otimes p_2 \ \equiv\ p_2,$ if $[l_1, u_1] = [1, 1]$ and $KB \cap \{a \wedge b \rightarrow \bot,\, a \rightarrow b\} = \emptyset$ $p_1 \otimes p_2 \ \equiv\ p_1,$ if $[l_2, u_2] = [1, 1]$ and $KB \cap \{a \wedge b \rightarrow \bot,\, b \rightarrow a\} = \emptyset$ $p_1 \oplus p_2 \ \equiv\ p_2,$ if $[l_1, u_1] = [0, 0]$ and $KB \cap \{\top \rightarrow a \vee b,\, b \rightarrow a\} = \emptyset$ $p_1 \oplus p_2 \ \equiv\ p_1,$ if $[l_2, u_2] = [0, 0]$ and $KB \cap \{\top \rightarrow a \vee b,\, a \rightarrow b\} = \emptyset$ $p_1 \ominus p_2 \ \equiv\ p_1,$ if $[l_2, u_2] = [0, 0]$ and $KB \cap \{\top \rightarrow a \vee b,\, a \rightarrow b\} = \emptyset$
ANNIHILATOR	$p_1 \otimes p_2 \ \equiv\ \bot,$ if $[l_1, u_1] = [0, 0]$ or $[l_2, u_2] = [0, 0]$ $p_1 \oplus p_2 \ \equiv\ \top,$ if $[l_1, u_1] = [1, 1]$ or $[l_2, u_2] = [1, 1]$ $p_1 \ominus p_2 \ \equiv\ \bot,$ if $[l_1, u_1] = [0, 0]$ or $[l_2, u_2] = [1, 1]$

certain restrictions as neutral elements like 0 and 1 in Boolean algebra. The obvious question is whether the conditions on KB in case of WEAK IDENTITY are really necessary. As the following examples show, this is true. Note that the examples just refer to the second, the fourth, and the fifth entry of WEAK IDENTITY, as the first (resp., third) entry is symmetric to the second (resp., fourth) one.

Example 2.13 a) Consider the static probabilistic conjunction strategies $\otimes_{me}$ and $\otimes_{li}$ for mutual exclusion and left implication, respectively. That is, let $KB = \{a \wedge b \rightarrow \bot\}$ and $KB = \{b \rightarrow a\}$, respectively, where a and b are two distinct basic events. Moreover, consider the probabilistic pairs $p_1 = (e_1, [0, 1])$ and $p_2 = (e_2, [1, 1])$.

It is now easy to verify that $KB \cup \{(a, [0, 1]), (b, [1, 1])\}$ is satisfiable and that

$$p_1 \otimes_{me} p_2 \equiv \bot \not\equiv p_1 \quad \text{and} \quad p_1 \otimes_{li} p_2 \equiv \top \not\equiv p_1 \,.$$

b) Consider the static probabilistic disjunction strategies $\oplus_{ex}$ and $\oplus_{ri}$ for exhaustion and right implication, respectively. That is, let $KB = \{\top \rightarrow a \vee b\}$ and $KB = \{a \rightarrow b\}$, respectively, where a and b are two distinct basic events. Moreover, consider the probabilistic pairs $p_1 = (e_1, [0, 1])$ and $p_2 = (e_2, [0, 0])$.

It is now easy to verify that $KB \cup \{(a, [0, 1]), (b, [0, 0])\}$ is satisfiable and that

$$p_1 \oplus_{ex} p_2 \equiv \top \not\equiv p_1 \quad \text{and} \quad p_1 \oplus_{ri} p_2 \equiv \bot \not\equiv p_1 \,,$$
$$p_1 \ominus_{ex} p_2 \equiv \top \not\equiv p_1 \quad \text{and} \quad p_1 \ominus_{ri} p_2 \equiv \bot \not\equiv p_1 \,.$$

3 Data Model

We assume that $\mathcal{A}$ is a given nonempty set of *attribute names*, and $\mathcal{T}$ is a given nonempty set of *atomic types*, where every atomic type $T \in \mathcal{T}$ has an associated countable *domain* $\mathrm{dom}(T)$ of values.

We define *complex value types* (or simply *types*) inductively as follows:

- Every atomic type from $\mathcal{T}$ is a type.
- If T is a type, then the set $\{T\}$ is a type.
- If $A_1, \ldots, A_k$ with $k \geq 0$ are distinct attribute names from $\mathcal{A}$ and $T_1, \ldots, T_k$ are types, then the mapping $T = \{(A_1, T_1), \ldots, (A_k, T_k)\}$, also written $[A_1\colon T_1, \ldots, A_k\colon T_k]$, is a type (called *tuple type*). We call $A_1, \ldots, A_k$ the *top-level attribute names* of T, and use $T.A_i$ to denote T_i.

Complex values are now inductively defined as follows:

- If $T \in \mathcal{T}$ is an atomic type, then every $v \in \mathrm{dom}(T)$ is a complex value of type T.
- If $v_1, \ldots, v_k$ with $k \geq 0$ are complex values of type T, then the set $\{v_1, \ldots, v_k\}$ is a complex value of type $\{T\}$.
- If $A_1, \ldots, A_k$ with $k \geq 0$ are distinct attribute names from $\mathcal{A}$ and $v_1, \ldots, v_k$ are complex values of types $T_1, \ldots, T_k$, then the mapping $\{(A_1, v_1), \ldots, (A_k, v_k)\}$, also written $[A_1\colon v_1, \ldots, A_k\colon v_k]$, is a complex value of type $[A_1\colon T_1, \ldots, A_k\colon T_k]$.

We use $\mathrm{dom}(T)$ to denote the set of all complex values v of a type T.

Given a tuple type T, a *complex value tuple* (*cv-tuple*) of type T is simply a complex value of type T. For cv-tuples $v = [A_1\colon v_1, \ldots, A_k\colon v_k]$, we use $v.A_i$ to denote v_i. A *complex value relation* (*cv-relation*) of type T is a finite set of cv-tuples of type T.

A *probabilistic complex value tuple* (*pcv-tuple*) of type T is a mapping $t = \{(\mathsf{data}, v), (\mathsf{lb}, r), (\mathsf{ub}, s), (\mathsf{path}, e)\}$, also written $[\mathsf{data}\colon v,\ \mathsf{lb}\colon r,\ \mathsf{ub}\colon s,\ \mathsf{path}\colon e]$, where v is a cv-tuple of type T, r and s are rational numbers with $0 \leq r \leq s \leq 1$, and e is an event.

We use $t.\mathsf{data}$, $t.\mathsf{lb}$, $t.\mathsf{ub}$, $t.\mathsf{path}$, and $t.\mathsf{prob}$ to denote v, r, s, e, and $(e, [r, s])$, respectively.

Definition 3.1 A *probabilistic complex value relation* (*pcv-relation*) of type T is a finite set of pcv-tuples of type T. A *base pcv-relation* is a pcv-relation r such that (i) $t.\mathsf{path} \in \mathcal{B}$ for all $t \in r$ and (ii) $t_1.\mathsf{path} \neq t_2.\mathsf{path}$ for all $t_1, t_2 \in r$ with $t_1.\mathsf{data} \neq t_2.\mathsf{data}$. A *probabilistic complex value database* (*pcv-database*) is a finite set of base pcv-relations that are defined over pairwise disjoint sets of basic events.

A pcv-relation r is called *compact* iff $t_1.\mathsf{prob} = t_2.\mathsf{prob}$ for all $t_1, t_2 \in r$ with $t_1.\mathsf{data} = t_2.\mathsf{data}$. In the sequel, we identify compact pcv-relations r of type R with pairs (δ, μ), where δ is a cv-relation of type R and μ is a mapping from δ to the set of all probabilistic pairs. Each non-compact pcv-relation can be transformed into a compact pcv-relation by applying a compaction operation.

Definition 3.2 Given a pcv-relation r and a commutative and associative probabilistic disjunction strategy $\oplus$ that has the identity property, the *compaction of r under* $\oplus$, denoted $\kappa^{\oplus}(r)$, is defined as the pcv-relation (δ, μ), where:

$$\delta = \{w.\mathsf{data} \mid w \in r\}, \text{ and}$$

$$\mu(v) = \textstyle\bigoplus_{w \in r,\, w.\mathsf{data} = v} w.\mathsf{prob} \ \text{ for all } v \in \delta.$$

In the sequel, we thus assume that all pcv-relations are compact. Unless stated otherwise, we also assume that $t.\mathsf{prob} \not\equiv \bot$ for all tuples t in a pcv-relation r. That is, we make the closed world assumption that $t.\mathsf{prob} \equiv \bot$ for all $t \notin r$. Notice that this assumption can be overridden for any value v by adding a tuple t with $t.\mathsf{data} = v$ and $t.\mathsf{prob} = [0, 1]$.

Example 3.3 A compact pcv-relation is shown in Table 3. It contains information about chemical experiments in which several substances are checked for a certain ingredient. Each experiment has information about the substance analyzed (substance-id), the laboratory (lab), the date (date), the assistants who conducted the experiment (assistants), and the result (α-result).

The probabilistic information in this pcv-relation can be interpreted as subjective belief of an agent. Each pcv-tuple t in a relation r then means that the probability that membership of t.data in r holds in the real world ranges from t.lb to t.ub. For example, the first pcv-tuple t could say that the probability that substance S89 was analyzed in laboratory L17 on 02/17/99 by Jack and Jill with positive α-result lies between 0.7 and 0.8 (this could mean that we are unsure about the whole information in t.data or that we are just unsure about the α-result). Another possible interpretation is that the probability that the positive α-result describes correctly the properties of substance S89 lies between 0.7 and 0.8.

Table 3: A compact pcv-relation

<table>
<tr><td colspan="5">data</td><td>lb</td><td>ub</td><td>path</td></tr>
<tr><td>substance-id</td><td colspan="3">α-arrangement</td><td>α-result</td><td rowspan="3">0.7</td><td rowspan="3">0.8</td><td rowspan="3">e_1</td></tr>
<tr><td rowspan="2">S89</td><td>lab</td><td>date</td><td>assistants</td><td rowspan="2">$+$</td></tr>
<tr><td>L17</td><td>02/17/99</td><td>$\{$Jack, Jill$\}$</td></tr>
<tr><td>substance-id</td><td colspan="3">α-arrangement</td><td>α-result</td><td rowspan="3">0.6</td><td rowspan="3">0.9</td><td rowspan="3">e_2</td></tr>
<tr><td rowspan="2">S89</td><td>lab</td><td>date</td><td>assistants</td><td rowspan="2">$+$</td></tr>
<tr><td>L10</td><td>02/13/99</td><td>$\{$Joe, Jim$\}$</td></tr>
<tr><td>substance-id</td><td colspan="3">α-arrangement</td><td>α-result</td><td rowspan="3">0.5</td><td rowspan="3">0.7</td><td rowspan="3">e_3</td></tr>
<tr><td rowspan="2">S64</td><td>lab</td><td>date</td><td>assistants</td><td rowspan="2">$-$</td></tr>
<tr><td>L12</td><td>02/17/99</td><td>$\{$Janet, Jill$\}$</td></tr>
</table>

4 Algebra

In this section, we present an algebra on probabilistic complex value relations, which can be used for querying probabilistic complex value databases. The algebra is an extension of the familiar relational algebra for relational databases, whose main operations are selection, projection, renaming join, Cartesian product, and the standard set operations of union, intersection, and difference. We define these operations for our model of pcv-databases, where we group operators into subsections.

4.1 Selection, Projection, and Renaming

Selection. We start with the selection operation on pcv-relations. For that, we need to formally define the syntax and semantics of selection conditions.

We first define their syntax, and define *terms* inductively as follows. A term is a complex value v, the tuple variable x, or an expression $t.A$, where t is a term and $A \in \mathcal{A} \cup \{\mathsf{data}, \mathsf{lb}, \mathsf{ub}\}$. The tuple variable x ranges over tuples in pcv-relation of appropriate type, which is given implicitly by the relation on which the selection condition is applied.

We define *selection conditions* inductively as follows: (i) If t_1 and t_2 are terms and θ is from $\{=, \leq, \in, \subseteq\}$, then $t_1\,\theta\,t_2$ is an *atomic* selection condition. (ii) If ϕ_1 and ϕ_2 are selection conditions, then $\neg\phi_1$ and $(\phi_1 \wedge \phi_2)$ are selection conditions. We use $(\phi_1 \vee \phi_2)$ to abbreviate $\neg(\neg\phi_1 \wedge \neg\phi_2)$.

The semantics of selection conditions is as follows. The *interpretation* of a term t in a pcv-tuple w, denoted $[t]_w$, is inductively defined by $[v]_w = v$, $[x]_w = w$, and $[t.A]_w = [t]_w.A$ if $[t]_w.A$ is defined (i.e., the attribute A does exist for $[t]_w.A$).

A selection condition ϕ is *applicable* to a tuple type T iff for all atomic selection conditions $t_1\,\theta\,t_2$ that occur in ϕ and all pcv-tuples w of type T the values of $[t_1]_w$, $[t_2]_w$, and $[t_1]_w\,\theta\,[t_2]_w$ are defined. A selection condition ϕ is *probability-free* iff it does not contain the expressions lb and ub.

For selection conditions ϕ that are applicable to T, the *satisfaction* of ϕ in a pcv-tuple w of type T, denoted $w \models \phi$, is inductively defined as follows:

- $w \models t_1\,\theta\,t_2$ iff $[t_1]_w\,\theta\,[t_2]_w$,
- $w \models \neg\phi$ iff not $w \models \phi$,
- $w \models \phi \wedge \psi$ iff $w \models \phi$ and $w \models \psi$.

For complex values v of type T and probability-free selection conditions ϕ that are applicable to T, we use $v \models \phi$ to express that $[\mathsf{data}\colon v,\ \mathsf{lb}\colon l,\ \mathsf{ub}\colon u,\ \mathsf{path}\colon e] \models \phi$ for some probabilistic pair $(e, [l, u])$.

Definition 4.1 Let $r = (\delta, \mu)$ be a pcv-relation of type R and let ϕ be a selection condition that is applicable to R. The *selection on r with respect to ϕ*, denoted $\sigma_\phi(r)$, is the pcv-relation (δ', μ') of type R, where:

$$\delta' \;=\; \{w.\mathsf{data} \mid w \in r,\ w \models \phi\},$$

$$\mu'(v) \;=\; \mu(v) \text{ for all } v \in \delta'.$$

Example 4.2 Let us take the pcv-relation given in Table 3. Let the selection condition ϕ be defined by

$$\phi \;=\; ((\,x.\mathsf{lb} \geq 0.7 \;\vee\; x.\mathsf{data}.\alpha\text{-}\mathsf{result} = -) \wedge \\ \mathsf{Jack} \in x.\mathsf{data}.\alpha\text{-}\mathsf{arrangement.assistants}\,)\,.$$

The result of the selection operation with respect to ϕ is shown in Table 4.

Table 4: Result of selection

data					lb	ub	path
substance-id	α-arrangement			α-result			
	lab	date	assistants				
S89	L17	02/17/99	{Jack, Jill}	+	0.7	0.8	e_1

Projection. We next define projection on pcv-relations. We define the *subtype* relationship $T_1 \preceq T_2$ on all types inductively as follows: (i) $T \preceq T$ for any atomic type $T \in \mathcal{T}$. (ii) If $T_1 \preceq T_2$, then $\{T_1\} \preceq \{T_2\}$. (iii) If $A_1, \ldots, A_k$ with $k \geq 0$ are distinct attribute names from $\mathcal{A}$ and $T_1, \ldots, T_l, S_1, \ldots, S_k$ with $l \leq k$ are types such that $T_i \preceq S_i$ for $i = 1, \ldots, l$, then $[A_1\colon T_1, \ldots, A_l\colon T_l] \preceq [A_1\colon S_1, \ldots, A_k\colon S_k]$.

Let v be a complex value of type T and let S be a subtype of T. The *projection of v to S*, denoted $\pi_S(v)$, is inductively defined as follows:

- If S is an atomic type, then $\pi_S(v) = v$.
- If $v = \{v_1, \ldots, v_k\}$ and $S = \{S'\}$, then $\pi_S(v) = \{\pi_{S'}(v_1), \ldots, \pi_{S'}(v_k)\}$.
- If $v = [A_1\colon v_1, \ldots, A_k\colon v_k]$ and $S = [A_1\colon S_1, \ldots, A_l\colon S_l]$, then $\pi_S(v) = [A_1\colon \pi_{S_1}(v_1), \ldots, A_l\colon \pi_{S_l}(v_l)]$.

Let r be a cv-relation of type R and let S be a subtype of R. The *projection of r to S*, denoted $\pi_S(r)$, is the cv-relation $\{\pi_S(t) \mid t \in r\}$ of type S.

Definition 4.3 Let $r = (\delta, \mu)$ be a pcv-relation of type R, let S be a subtype of R, and let $\oplus$ be a commutative and associative disjunction strategy that has the identity property. The *projection of r to S under $\oplus$*, denoted $\pi_S^{\oplus}(r)$, is defined as the pcv-relation (δ', μ') of type S, where:

$$\delta' = v \in \pi_S(\delta) \mid \bigoplus_{w \in \delta,\, \pi_S(w)=v} \mu(w) \not\equiv \bot\},$$

$$\mu'(v) = \bigoplus_{w \in \delta,\, \pi_S(w)=v} \mu(w) \text{ for all } v \in \delta'.$$

Example 4.4 The projection of the pcv-relation from Table 3 to a tuple type over the attribute names substance-id and α-result under $\oplus_{pc}$ is given in Table 5.

Table 5: Result of projection

data		lb	ub	path
substance-id	α-result	0.7	0.9	$e_1 \vee e_2$
S89	$+$			
substance-id	α-result	0.5	0.7	e_3
S64	$-$			

Renaming. Let T be a type and let $\mathbf{A}$ denote the set of all attribute names that occur in T. A *renaming condition* for T is an expression $B_1, \ldots, B_l \leftarrow C_1, \ldots, C_l$, where $B_1, \ldots, B_l$ is a sequence of distinct attribute names from $\mathcal{A}$ and $C_1, \ldots, C_l$ is a sequence of distinct attribute names from $(\mathcal{A} - \mathbf{A}) \cup \{B_1, \ldots, B_l\}$.

Let T be a type, let v be a complex value of type T, and let $N = B_1, \ldots, B_l \leftarrow C_1, \ldots, C_l$ be a renaming condition for T. The *renaming of T (resp., v) with respect to N*, denoted $\rho_N(T)$ (resp., $\rho_N(v)$), is obtained from T (resp., v) by replacing each attribute name B_i with $i \in [1:l]$ by the attribute name C_i. Observe that ρ_N is an injective mapping, and thus the inverse mapping ρ_N^{-1} exists.

Let r be a cv-relation of type R and let N be a renaming condition for R. The *renaming of r with respect to N*, denoted $\rho_N(r)$, is the cv-relation $\{\rho_N(t) \mid t \in r\}$ of type $\rho_N(R)$.

Definition 4.5 Let $r = (\delta, \mu)$ be a pcv-relation of type R and let N be a renaming condition for R. The *renaming of r with respect to N*, denoted $\rho_N(r)$, is the pcv-relation (δ', μ') of type $\rho_N(R)$, where:

$$\delta' = \rho_N(\delta),$$

$$\mu'(v) = \mu(\rho_N^{-1}(v)) \text{ for all } v \in \delta'.$$

4.2 Join and Cartesian Product

Recall that in relational algebra, the join $r \bowtie s$ of relations r and s returns a relation which contains all combinations of tuples t_1 and t_2 from r and s, respectively, that have the same value on their common attributes. Cartesian product $r \times s$ occurs as the special case in which r and s have no common attributes. Different from selection, lifting these operations to pcv-relations involves computation of new probabilistic values.

Join. Two tuple types T_1 and T_2 over the sets of top-level attribute names $\mathbf{A}_2$ and $\mathbf{A}_2$, respectively, are *join-compatible* iff $T_1.A = T_2.A$ for all $A \in \mathbf{A}_1 \cap \mathbf{A}_2$. The *join* of two such types T_1 and T_2, denoted $T_1 \bowtie T_2$, is the tuple type T over $\mathbf{A}_1 \cup \mathbf{A}_2$, where $T.A = T_1.A$ if $A \in \mathbf{A}_1$ and $T.A = T_2.A$ if $A \in \mathbf{A}_2$.

Let v_1 and v_2 be two cv-tuples of join-compatible types T_1 and T_2 over $\mathbf{A}_1$ and $\mathbf{A}_2$, respectively, with $v_1.A = v_2.A$ for all $A \in \mathbf{A}_1 \cap \mathbf{A}_2$. The *join* of v_1 and v_2, denoted $v_1 \bowtie v_2$, is defined as the cv-tuple v of type $T_1 \bowtie T_2$, where $v.A = v_1.A$ if $A \in \mathbf{A}_1$ and $v.A = v_2.A$ if $A \in \mathbf{A}_2$.

Let r_1 and r_2 be two cv-relations of join-compatible types R_1 and R_2 over $\mathbf{A}_1$ and $\mathbf{A}_2$, respectively. The *join of* r_1 *and* r_2, denoted $r_1 \bowtie r_2$, is the cv-relation r of type $R_1 \bowtie R_2$ with $r = \{t_1 \bowtie t_2 \mid t_1 \in r_1, t_2 \in r_2, t_1.A = t_2.A \text{ for all } A \in \mathbf{A}_1 \cap \mathbf{A}_2\}$.

Definition 4.6 Let $r_1 = (\delta_1, \mu_1)$ and $r_2 = (\delta_2, \mu_2)$ be two pcv-relations of join-compatible types R_1 and R_2, respectively, and let $\otimes$ be a probabilistic conjunction strategy. The *join of* r_1 *and* r_2 *under* $\otimes$, denoted $r_1 \bowtie^{\otimes} r_2$, is defined as the pcv-relation (δ, μ) of type $R_1 \bowtie R_2$, where:

$$\delta = \{v \in \delta_1 \bowtie \delta_2 \mid \mu_1(\pi_{R_1}(v)) \otimes \mu_2(\pi_{R_2}(v)) \not\equiv \bot\},$$

$$\mu(v) = \mu_1(\pi_{R_1}(v)) \otimes \mu_2(\pi_{R_2}(v)) \text{ for all } v \in \delta.$$

Example 4.7 The join of the pcv-relations from Tables 5 and 6 under the probabilistic conjunction strategy $\otimes_{in}$ is given in Table 7.

Table 6: Second input pcv-relation of join

<table>
<tr><th colspan="2">data</th><th>lb</th><th>ub</th><th>path</th></tr>
<tr><td>substance-id</td><td>β-result</td><td rowspan="2">0.8</td><td rowspan="2">0.8</td><td rowspan="2">e_4</td></tr>
<tr><td>S89</td><td>$-$</td></tr>
<tr><td>substance-id</td><td>β-result</td><td rowspan="2">0.8</td><td rowspan="2">1.0</td><td rowspan="2">e_5</td></tr>
<tr><td>S37</td><td>$+$</td></tr>
</table>

Table 7: Result of join

<table>
<tr><th colspan="3">data</th><th>lb</th><th>ub</th><th>path</th></tr>
<tr><td>substance-id</td><td>α-result</td><td>β-result</td><td rowspan="2">0.56</td><td rowspan="2">0.72</td><td rowspan="2">$(e_1 \vee e_2) \wedge e_4$</td></tr>
<tr><td>S89</td><td>$+$</td><td>$-$</td></tr>
</table>

Cartesian Product. We next define the Cartesian product as a special case of join. Two tuple types T_1 and T_2 over the sets of top-level attribute names $\mathbf{A}_1$ and $\mathbf{A}_2$, respectively, are *Cartesian-product-compatible* iff $\mathbf{A}_1$ and $\mathbf{A}_2$ are disjoint. The *Cartesian product* of such types T_1 and T_2, denoted $T_1 \times T_2$, is the tuple type $T_1 \bowtie T_2$.

Definition 4.8 Let r_1 and r_2 be two pcv-relations of Cartesian-product-compatible types R_1 and R_2, respectively, and let $\otimes$ be a probabilistic conjunction strategy. The *Cartesian product of* r_1 *and* r_2 *under* $\otimes$, denoted $r_1 \times^{\otimes} r_2$, is the pcv-relation $r_1 \bowtie^{\otimes} r_2$ of type $R_1 \times R_2$.

4.3 Union, Intersection, and Difference

We now define the set operations of union, intersection, and difference on pcv-relations.

Definition 4.9 Let $r_1 = (\delta_1, \mu_1)$ and $r_2 = (\delta_2, \mu_2)$ be two pcv-relations of the same type R. Let $\oplus$ be a probabilistic disjunction strategy. The *union of* r_1 *and* r_2 *under* $\oplus$, denoted $r_1 \cup^{\oplus} r_2$, is the pcv-relation (δ, μ) of type R, where:

$$\begin{aligned}\delta \;&=\; \{v \in \delta_1 - \delta_2 \mid \mu_1(v) \oplus \bot \not\equiv \bot\} \,\cup \\ &\quad\; \{v \in \delta_2 - \delta_1 \mid \bot \oplus \mu_2(v) \not\equiv \bot\} \,\cup\, \{v \in \delta_1 \cap \delta_2 \mid \mu_1(v) \oplus \mu_2(v) \not\equiv \bot\},\end{aligned}$$

$$\mu(v) \;=\; \begin{cases} \mu_1(v) \oplus \bot & \text{if } v \in \delta_1 - \delta_2 \\ \bot \oplus \mu_2(v) & \text{if } v \in \delta_2 - \delta_1 \\ \mu_1(v) \oplus \mu_2(v) & \text{if } v \in \delta_1 \cap \delta_2\,. \end{cases}$$

Example 4.10 The union of the pcv-relations from Tables 5 and 8 under the probabilistic disjunction strategy $\oplus_{pc}$ is given in Table 9.

Table 8: Second input pcv-relation of union

data		lb	ub	path
substance-id	α-result	0.2	0.3	e_6
S89	$+$			
substance-id	α-result	0.3	0.4	e_7
S37	$-$			

Definition 4.11 Let $r_1 = (\delta_1, \mu_1)$ and $r_2 = (\delta_2, \mu_2)$ be two pcv-relations of the same type R. Let $\otimes$ be a probabilistic conjunction strategy. The *intersection of* r_1 *and* r_2 *under* $\otimes$, denoted $r_1 \cap^{\otimes} r_2$, is the pcv-relation (δ, μ) of type R, where:

$$\begin{aligned}\delta \;&=\; \{v \in \delta_1 \cap \delta_2 \mid \mu_1(v) \otimes \mu_2(v) \not\equiv \bot\}, \\ \mu(v) \;&=\; \mu_1(v) \otimes \mu_2(v).\end{aligned}$$

Definition 4.12 Let $r_1 = (\delta_1, \mu_1)$ and $r_2 = (\delta_2, \mu_2)$ be two pcv-relations of the same type R. Let $\ominus$ be a probabilistic difference strategy. The *difference of* r_1 *and* r_2 *under* $\ominus$, denoted $r_1 -^{\ominus} r_2$, is the pcv-relation (δ, μ) of type R, where:

Table 9: Result of union

<table>
<tr><td colspan="2">data</td><td>lb</td><td>ub</td><td>path</td></tr>
<tr><td>substance-id</td><td>α-result</td><td rowspan="2">0.7</td><td rowspan="2">0.9</td><td rowspan="2">$e_1 \vee e_2 \vee e_6$</td></tr>
<tr><td>S89</td><td>$+$</td></tr>
<tr><td>substance-id</td><td>α-result</td><td rowspan="2">0.3</td><td rowspan="2">0.4</td><td rowspan="2">e_7</td></tr>
<tr><td>S37</td><td>$-$</td></tr>
<tr><td>substance-id</td><td>α-result</td><td rowspan="2">0.5</td><td rowspan="2">0.7</td><td rowspan="2">e_3</td></tr>
<tr><td>S64</td><td>$-$</td></tr>
</table>

$$\delta = \{v \in \delta_1 - \delta_2 \mid \mu_1(v) \ominus \bot \not\equiv \bot\} \cup \{v \in \delta_1 \cap \delta_2 \mid \mu_1(v) \ominus \mu_2(v) \not\equiv \bot\},$$

$$\mu(v) = \begin{cases} \mu_1(v) \ominus \bot & \text{if } v \in \delta_1 - \delta_2 \\ \mu_1(v) \ominus \mu_2(v) & \text{if } v \in \delta_1 \cap \delta_2 \,. \end{cases}$$

4.4 Relationship to Classical Relational Algebra

We now show that pcv-relations and the algebra on pcv-relations properly extend classical relations and the classical relational algebra. We first briefly recall classical relations and the classical relational algebra.

A classical *relation scheme* R is a set of attribute names $\{A_1, \ldots, A_k\}$ with $k \geq 0$, where each A_i with $i \in \{1, \ldots, k\}$ is assigned a domain $\mathrm{dom}(A_i)$. A classical *relation* r over R is a finite set of mappings v (called *tuples*) from R to $\bigcup\{\mathrm{dom}(A_i) \mid i \in \{1, \ldots, k\}\}$ such that $v(A_i) \in \mathrm{dom}(A_i)$ for all $i \in \{1, \ldots, k\}$.

Clearly, every classical relation r over scheme $R = \{A_1, \ldots, A_k\}$ corresponds to a cv-relation of type $\varepsilon(R) = [A_1 \colon T_1, \ldots, A_k \colon T_k]$, where each T_i with $i \in \{1, \ldots, k\}$ is an atomic type with domain $\mathrm{dom}(T_i) = \mathrm{dom}(A_i)$. The operations in Sections 4.1–4.3 (selection, projection, renaming, join, Cartesian product, union, intersection and difference) are defined for classical relations through the data part δ of the respective definitions for pcv-relations, if all conditions involving probabilistic values μ are replaced by true except $\mu_1(v) \ominus \mu_2(v) \not\equiv \bot$ in the definition of difference, which is replaced by false. Selection conditions are syntactically restricted such that comparison $t_1 \; \theta \; t_2$ of terms t_1 and t_2 is only defined for θ from $\{=, \leq\}$. We refer to Appendix C for the details.

The correspondence ε naturally extends to the instance level by mapping each tuple $v = \{(A_1, v_1), \ldots, (A_k, v_k)\}$ over R to the pcv-tuple $\varepsilon(v) = [\mathsf{data}\colon [A_1 \colon v_1, \ldots, A_k \colon v_k], \mathsf{lb}\colon 1, \mathsf{ub}\colon 1, \mathsf{path}\colon e_v]$ of type $\varepsilon(R)$, where e_v is a basic event from $\mathcal{B}$, and each classical base relation r over R to the base pcv-relation $\varepsilon(r) = \{\varepsilon(v) \mid v \in r\}$ of type $\varepsilon(R)$ (which clearly satisfies $t_1.\mathsf{path} \neq t_2.\mathsf{path}$ for all distinct $t_1, t_2 \in \varepsilon(r)$, if we assume that $e_{v_1} \neq e_{v_2}$ for all distinct $v_1, v_2 \in r$).

To show that our algebra properly extends the classical relational algebra, it remains to show that the operations in classical relational algebra commute with the correspondence ε. That is, for example, the result of classical selection $\sigma_\phi(r)$ on r must correspond to the result of selection $\sigma_{\varepsilon(\phi)}(\varepsilon(r))$ in our algebra on the corresponding pcv-relation $\varepsilon(r)$, where $\varepsilon(\phi)$ is the selection condition corresponding to ϕ, which is obtained from ϕ by replacing each occurrence of an attribute name A by $x.\mathsf{data}.A$. More formally, the equation $\varepsilon(\sigma_\phi(r)) = \sigma_{\varepsilon(\phi)}(\varepsilon(r))$ must hold. Similar equations, which are summarized in Theorem 4.14 below, must hold for the other operations.

To establish this, we first observe that each basic event b in a probabilistic pair $(b, [1, 1])$ is logically equivalent to $\top$. That is, we can tacitly assume that $t.\mathsf{prob} = \top$ for all pcv-tuples t in base pcv-relations $\varepsilon(r)$. The following lemma shows how the probabilistic pairs $\bot$ and $\top$ combine under different static probabilistic strategies.

Lemma 4.13 Let $\otimes$ (resp., $\oplus$, $\ominus$) be the static probabilistic conjunction (resp., disjunction, difference) strategy for any case among ignorance, independence, positive correlation, and negative correlation. Then, the result of applying $\otimes$, $\oplus$, and $\ominus$ to the probabilistic pairs $\bot$ and $\top$ is given in Figure 2.

$\otimes$	$\bot$	$\top$
$\bot$	$\bot$	$\bot$
$\top$	$\bot$	$\top$

$\oplus$	$\bot$	$\top$
$\bot$	$\bot$	$\top$
$\top$	$\top$	$\top$

$\ominus$	$\bot$	$\top$
$\bot$	$\bot$	$\bot$
$\top$	$\top$	$\bot$

Figure 2: Probabilistic combinations of $\bot$ and $\top$

We now show that the operations of classical relational algebra commute with ε under the probabilistic combination strategies in this lemma, which gives the desired result.

Theorem 4.14 Let r_1 and r_2 be two relations over the relation schemes R_1 and R_2, respectively, which are disjoint in Eq. (4.5) and identical in Eq. (4.6)–(4.8). Let ϕ be a selection condition applicable to R_1, let T be a relation scheme $T \subseteq R_1$, and let N be a renaming condition for R_1. Suppose that $\otimes$ (resp., $\oplus$, $\ominus$) is the static probabilistic conjunction (resp., disjunction, difference) strategy for any case among ignorance, independence, positive correlation, and negative correlation. Then,

$$\begin{aligned}
\sigma_{\varepsilon(\phi)}(\varepsilon(r_1)) &= \varepsilon(\sigma_\phi(r_1)) && (4.1)\\
\pi^{\oplus}_{\varepsilon(T)}(\varepsilon(r_1)) &= \varepsilon(\pi_T(r_1)) && (4.2)\\
\delta_N(\varepsilon(r_1)) &= \varepsilon(\delta_N(r_1)) && (4.3)\\
\varepsilon(r_1) \bowtie^{\otimes} \varepsilon(r_2) &= \varepsilon(r_1 \bowtie r_2) && (4.4)\\
\varepsilon(r_1) \times^{\otimes} \varepsilon(r_2) &= \varepsilon(r_1 \times r_2) && (4.5)\\
\varepsilon(r_1) \cup^{\oplus} \varepsilon(r_2) &= \varepsilon(r_1 \cup r_2) && (4.6)\\
\varepsilon(r_1) \cap^{\otimes} \varepsilon(r_2) &= \varepsilon(r_1 \cap r_2) && (4.7)\\
\varepsilon(r_1) -^{\ominus} \varepsilon(r_2) &= \varepsilon(r_1 - r_2)\,. && (4.8)
\end{aligned}$$

4.5 Computational Complexity

We now analyze the complexity of the introduced algebraic operations on pcv-relations. In particular, we show that each operation under static probabilistic combination strategies can be done in polynomial time in the size of the input relations.

We first provide some complexity results on probabilistic reasoning with probabilistic pairs under static dependence. The following theorem shows that the satisfiability check under static dependence is possible in constant time[1]. Furthermore, it shows that every application of a static probabilistic conjunction (resp., disjunction, difference) strategy can be done in constant time. Note that the proof of this theorem contains effective algorithms, which can be easily implemented.

Theorem 4.15 Let $KB \subseteq KB^{\star}(a, b)$ be a dependence information on two distinct basic events a and b. Then, the following holds.

(a) Given two probabilistic pairs $(e_1, [l_1, u_1])$ and $(e_2, [l_2, u_2])$ such that $l_1 \leq u_1$ and $l_2 \leq u_2$, deciding whether $KB \cup \{(a, [l_1, u_1]), (b, [l_2, u_2])\}$ is satisfiable can be done in time $O(1)$.

(b) Given two probabilistic pairs $(e_1, [l_1, u_1])$ and $(e_2, [l_2, u_2])$ such that $KB \cup \{(a, [l_1, u_1]), (b, [l_2, u_2])\}$ is satisfiable, computing $(a, [l_1, u_1]) \odot (b, [l_2, u_2])$ with $\odot \in \{\otimes_{KB}, \oplus_{KB}, \ominus_{KB}\}$ is possible in time $O(1)$.

We are now ready to analyze the complexity of our algebraic operations under static dependence. The following result shows that, under static probabilistic combination strategies, every unary operation on pcv-relations can be done in polynomial time. In fact, the renaming operation can even be done in linear time. Moreover, the same applies to the selection operation, if we assume that sets of complex values are represented as sets of totally ordered elements or as bitvectors.

Theorem 4.16 Let $\oplus$ be a static probabilistic disjunction strategy that is commutative and associative and that has the identity property. Then, the following holds.

(a) Given a pcv-relation r of type R, the compaction $\kappa^{\oplus}(r)$ is possible in polynomial time in the input size of r.

(b) Let ϕ be a fixed selection condition. Then, given a pcv-relation r of type R such that ϕ is applicable to R, computing the selection $\sigma_{\phi}(r)$ is possible in linear time in the input size of r.

(c) Given a pcv-relation r of type R and a renaming condition N for R, computing the renaming $\sigma_{\phi}(r)$ is possible in linear time in the input size of r and N.

(d) Given a pcv-relation r of type R and a subtype S of R, computing the projection $\pi^{\oplus}_{S}(r)$ is possible in polynomial time in the input size of r.

The next result shows that, under static probabilistic combination strategies, every binary operation on pcv-relations can be done in polynomial time.

Theorem 4.17 For any static probabilistic conjunction, disjunction, and difference strategy $\otimes$, $\oplus$, and $\ominus$, respectively, the following holds.

(a) Given two pcv-relations r_1 and r_2 of join-compatible (resp., Cartesian-product-compatible) types R_1 and R_2, the join $r_1 \bowtie^{\otimes} r_2$ (resp., Cartesian product $r_1 \times^{\otimes} r_2$) can be computed in polynomial time in the input size of r_1 and r_2.

(b) Given two pcv-relations r_1 and r_2 of the same type R, the intersection $r_1 \cap^{\otimes} r_2$ (resp., union $r_1 \cup^{\oplus} r_2$, difference $r_1 -^{\ominus} r_2$) can be computed in polynomial time in the input size of r_1 and r_2.

[1]We assume a standard RAM machine model in which each standard arithmetic operation on integers (resp., rationals) takes one step.

Unfortunately, these tractability results generally do not carry over to the dynamic case. The next theorem shows that there, the satisfiability check is NP-complete.

Theorem 4.18 Given a dependence information $KB(a, b) \subseteq KB^\star(a, b)$ on two distinct basic events a, b and two probabilistic pairs $p_1 = (e_1, [l_1, u_1])$, $p_2 = (e_2, [l_2, u_2])$ where $l_1 \leq u_1$ and $l_2 \leq u_2$, deciding whether $KB(e_1, e_2) \cup \{p_1, p_2\}$ is satisfiable is NP-complete. Hardness holds even if $KB(a, b)$ is fixed to an arbitrary satisfiable dependence information.

In the restricted case in which only conjunctive (resp., disjunctive) events are involved, however, we also gain tractability under dynamic dependence. In detail, an event e is *conjunctive* (resp., *disjunctive*) iff e is either $\perp$, or $\top$, or a conjunction $a_1 \wedge \ldots \wedge a_k$ (resp., disjunction $a_1 \vee \ldots \vee a_k$) of basic events $a_1, \ldots, a_k$. A probabilistic pair $(e, [l, u])$ is *conjunctive* (resp., *disjunctive*) iff e is conjunctive (resp., disjunctive).

Theorem 4.19 Let $KB(a, b) \subseteq KB^\star(a, b)$ be a dependence information on two distinct basic events a and b. Then, the following holds.

(a) Given two conjunctive (resp., disjunctive) probabilistic pairs $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])$ with $l_1 \leq u_1$ and $l_2 \leq u_2$, deciding whether $KB(e_1, e_2) \cup \{p_1, p_2\}$ is satisfiable can be done in polynomial time in the input size of p_1 and p_2.

(b) Given two conjunctive (resp., disjunctive) probabilistic pairs $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])$ such that $KB(e_1, e_2) \cup \{p_1, p_2\}$ is satisfiable, computing $p_1 \odot p_2$ with $\odot \in \{\otimes_{KB(e_1, e_2)}, \oplus_{KB(e_1, e_2)}, \ominus_{KB(e_1, e_2)}\}$ can be done in polynomial time in the input size of p_1 and p_2.

Hence, the tractability results in Theorems 4.16 and 4.17 carry over to the dynamic case in which all events in the input pcv-relations are either conjunctive or disjunctive.

5 Query Equivalences

We now focus on query equivalences. We assume that all involved conjunction and disjunction strategies are commutative and associative. Moreover, we assume that all disjunction strategies have the identity property. Any such strategies are called *proper*.

Our first result shows that the join operation can be reduced to renaming, Cartesian product, selection, and projection like in the classical relational algebra.

Theorem 5.1 Let r_1 and r_2 be two pcv-relations of join-compatible types R_1 and R_2 with sets of top-level attribute names $\mathbf{A}_1$ and $\mathbf{A}_2$, respectively. Let ρ_N replace each $A \in \mathbf{A}_1 \cap \mathbf{A}_2$ by the new attribute A', and let ϕ be the conjunction of all $x.A' = x.A$. Let $R = R_1 \bowtie R_2$. Then, for any proper probabilistic conjunction and disjunction strategies $\otimes$ and $\oplus$, respectively,

$$r_1 \bowtie^{\otimes} r_2 \quad = \quad \pi^{\oplus}_{R}(\sigma_\phi(\rho_N(r_1) \times^{\otimes} r_2)) \,. \tag{5.1}$$

We next show that most other query equivalences of the classical relational algebra also carry over to our algebra on pcv-relations. The following theorem shows that selections with respect to conjunctive selection conditions can be decomposed, and that sequences of selections can be reordered.

Theorem 5.2 Let r be a pcv-relation of type R, and let ϕ_1 and ϕ_2 be selection conditions applicable to R. Then,

$$\sigma_{\phi_1 \wedge \phi_2}(r) \quad = \quad \sigma_{\phi_1}(\sigma_{\phi_2}(r)) \tag{5.2}$$

$$\sigma_{\phi_1}(\sigma_{\phi_2}(r)) \quad = \quad \sigma_{\phi_2}(\sigma_{\phi_1}(r)) \,. \tag{5.3}$$

The following result shows that sequences of projections can be reordered and that certain selections can be pushed through projections.

Theorem 5.3 Let r be a pcv-relation of type R. Let S and T be types such that $S \preceq T \preceq R$, and let ϕ be a probability-free selection condition applicable to T. Then, for any proper probabilistic disjunction strategy $\oplus$,

$$\pi_S^{\oplus}(\pi_T^{\oplus}(r)) \quad = \quad \pi_S^{\oplus}(r) \tag{5.4}$$
$$\sigma_\phi(\pi_T^{\oplus}(r)) \quad = \quad \pi_T^{\oplus}(\sigma_\phi(r))\,. \tag{5.5}$$

The next result shows that selections and projections can be pushed through renaming operations.

Theorem 5.4 Let r be a pcv-relation of type R and let N be a renaming condition for R. Suppose that ϕ is a selection condition applicable to $\rho_N(R)$ and that $S \preceq \rho_N(R)$. Let $\phi' = \rho_N^{-1}(\phi)$ and $S' = \rho_N^{-1}(S)$. Then, for any proper probabilistic disjunction strategy $\oplus$,

$$\sigma_\phi(\rho_N(r)) \quad = \quad \rho_N(\sigma_{\phi'}(r)) \tag{5.6}$$
$$\pi_S^{\oplus}(\rho_N(r)) \quad = \quad \rho_N(\pi_{S'}^{\oplus}(r))\,. \tag{5.7}$$

The following theorem shows that joins are commutative and associative, and that certain selections and projections can be pushed through join operations.

Theorem 5.5 Let r_1, r_2, and r_3 be pcv-relations of pairwise join-compatible types R_1, R_2, and R_3, respectively. Let $S \preceq R_1 \bowtie R_2$ and let $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}$ be the sets of top-level attribute names of R_1, R_2, and S, respectively. For $i \in \{1,2\}$, let the tuple type S_i over $(\mathbf{A} \cup \mathbf{A}_{2-i+1}) \cap \mathbf{A}_i$ be such that $S_i.A = S.A$ for all $A \in (\mathbf{A} - \mathbf{A}_{2-i+1}) \cap \mathbf{A}_i$ and $S_i.A = R_i.A$ for all $A \in \mathbf{A}_1 \cap \mathbf{A}_2$. Finally, for $i \in \{1,2\}$, let ϕ_i be a probability-free selection condition applicable to R_i. Then, for any proper probabilistic conjunction and disjunction strategies $\otimes$ and $\oplus$, respectively, such that $\otimes$ is distributive over $\oplus$,

$$r_1 \bowtie^{\otimes} r_2 \quad = \quad r_2 \bowtie^{\otimes} r_1 \tag{5.8}$$
$$(r_1 \bowtie^{\otimes} r_2) \bowtie^{\otimes} r_3 \quad = \quad r_1 \bowtie^{\otimes} (r_2 \bowtie^{\otimes} r_3) \tag{5.9}$$
$$\sigma_{\phi_1 \wedge \phi_2}(r_1 \bowtie^{\otimes} r_2) \quad = \quad \sigma_{\phi_1}(r_1) \bowtie^{\otimes} \sigma_{\phi_2}(r_2) \tag{5.10}$$
$$\pi_S^{\oplus}(r_1 \bowtie^{\otimes} r_2) \quad = \quad \pi_S^{\oplus}(\pi_{S_1}^{\oplus}(r_1) \bowtie^{\otimes} \pi_{S_2}^{\oplus}(r_2))\,. \tag{5.11}$$

As Cartesian product is a special case of join, we get the following corollary.

Corollary 5.6 Let r_1, r_2, and r_3 be pcv-relations of pairwise Cartesian-product-compatible types R_1, R_2, and R_3, respectively. Let $S \preceq R_1 \times R_2$ and let $\mathbf{A}_1$, $\mathbf{A}_2$, and $\mathbf{A}$ be the sets of top-level attribute names of R_1, R_2, and S, respectively. Let for $i \in \{1,2\}$ the tuple type S_i over $\mathbf{A} \cap \mathbf{A}_i$ be such that $S_i.A = S.A$ for all $A \in \mathbf{A} \cap \mathbf{A}_i$. Finally, let ϕ_1 and ϕ_2 be two probability-free selection conditions that are applicable to R_1 and R_2, respectively. Then, for any proper probabilistic conjunction and disjunction strategies $\otimes$ and $\oplus$, respectively, such that $\otimes$ is distributive over $\oplus$,

$$r_1 \times^{\otimes} r_2 \quad = \quad r_2 \times^{\otimes} r_1 \tag{5.12}$$
$$(r_1 \times^{\otimes} r_2) \times^{\otimes} r_3 \quad = \quad r_1 \times^{\otimes} (r_2 \times^{\otimes} r_3) \tag{5.13}$$
$$\sigma_{\phi_1 \wedge \phi_2}(r_1 \times^{\otimes} r_2) \quad = \quad \sigma_{\phi_1}(r_1) \times^{\otimes} \sigma_{\phi_2}(r_2) \tag{5.14}$$
$$\pi_S^{\oplus}(r_1 \times^{\otimes} r_2) \quad = \quad \pi_{S_1}^{\oplus}(r_1) \times^{\otimes} \pi_{S_2}^{\oplus}(r_2)\,. \tag{5.15}$$

The next theorem shows that union and intersection operations are commutative and associative, and that certain selections can be pushed through union, intersection, and difference. Moreover, it shows that projections can be pushed through union, and that intersection is a special case of join.

Theorem 5.7 Let r_1, r_2, and r_3 be pcv-relations of type R, let ϕ be a probability-free selection condition that is applicable to R, and let $S \preceq R$. Then, for any proper probabilistic conjunction, disjunction, and difference strategies $\otimes$, $\oplus$, and $\ominus$, respectively,

$$r_1 \cup^{\oplus} r_2 = r_2 \cup^{\oplus} r_1 \tag{5.16}$$

$$(r_1 \cup^{\oplus} r_2) \cup^{\oplus} r_3 = r_1 \cup^{\oplus} (r_2 \cup^{\oplus} r_3) \tag{5.17}$$

$$r_1 \cap^{\otimes} r_2 = r_2 \cap^{\otimes} r_1 \tag{5.18}$$

$$(r_1 \cap^{\otimes} r_2) \cap^{\otimes} r_3 = r_1 \cap^{\otimes} (r_2 \cap^{\otimes} r_3) \tag{5.19}$$

$$\sigma_{\phi}(r_1 \cup^{\oplus} r_2) = \sigma_{\phi}(r_1) \cup^{\oplus} \sigma_{\phi}(r_2) \tag{5.20}$$

$$\sigma_{\phi}(r_1 \cap^{\otimes} r_2) = \sigma_{\phi}(r_1) \cap^{\otimes} \sigma_{\phi}(r_2) \tag{5.21}$$

$$\sigma_{\phi}(r_1 -^{\ominus} r_2) = \sigma_{\phi}(r_1) -^{\ominus} \sigma_{\phi}(r_2) \tag{5.22}$$

$$\pi^{\oplus}_{S}(r_1 \cup^{\oplus} r_2) = \pi^{\oplus}_{S}(r_1) \cup^{\oplus} \pi^{\oplus}_{S}(r_2) \tag{5.23}$$

$$r_1 \cap^{\otimes} r_2 = r_1 \bowtie^{\otimes} r_2\,. \tag{5.24}$$

Finally, the following theorem shows that certain selections involving probability bounds can also be pushed through join, Cartesian product, and intersection.

Theorem 5.8 Let r_1 and r_2 be two pcv-relations of types R_1 and R_2, respectively, where the types are join-compatible in Eq. (5.25), Cartesian-product-compatible in Eq. (5.26), and identical in Eq. (5.27). Let ϕ be a selection condition of the form $x.\mathsf{lb} \geq c$ or $x.\mathsf{ub} \geq c$ where $c \in [0, 1]$ is rational. Then, for any proper probabilistic conjunction strategy $\otimes$ that has the bottomline property,

$$\sigma_{\phi}(r_1 \bowtie^{\otimes} r_2) = \sigma_{\phi}(\sigma_{\phi}(r_1) \bowtie^{\otimes} \sigma_{\phi}(r_2)) \tag{5.25}$$

$$\sigma_{\phi}(r_1 \times^{\otimes} r_2) = \sigma_{\phi}(\sigma_{\phi}(r_1) \times^{\otimes} \sigma_{\phi}(r_2)) \tag{5.26}$$

$$\sigma_{\phi}(r_1 \cap^{\otimes} r_2) = \sigma_{\phi}(\sigma_{\phi}(r_1) \cap^{\otimes} \sigma_{\phi}(r_2))\,. \tag{5.27}$$

By Theorem 5.2 (5.2), Theorem 5.5 (5.10), Corollary 5.6 (5.14), and Theorem 5.7 (5.21), we get the following corollary.

Corollary 5.9 Let r_1 and r_2 be two pcv-relations of types R_1 and R_2, respectively, where the types are join-compatible in Eq. (5.28), Cartesian-product-compatible in Eq. (5.29), and identical in Eq. (5.30). Let ϕ be a selection condition of the form $x.\mathsf{lb} \geq c$ or $x.\mathsf{ub} \geq c$ where $c \in [0, 1]$ is rational. Finally, let ϕ_1 and ϕ_2 be probability-free selection conditions applicable to R_1 and R_2, respectively. Then, for any proper probabilistic conjunction strategy $\otimes$ that has the bottomline property,

$$\sigma_{\phi \wedge \phi_1 \wedge \phi_2}(r_1 \bowtie^{\otimes} r_2) = \sigma_{\phi}(\sigma_{\phi \wedge \phi_1}(r_1) \bowtie^{\otimes} \sigma_{\phi \wedge \phi_2}(r_2)) \tag{5.28}$$

$$\sigma_{\phi \wedge \phi_1 \wedge \phi_2}(r_1 \times^{\otimes} r_2) = \sigma_{\phi}(\sigma_{\phi \wedge \phi_1}(r_1) \times^{\otimes} \sigma_{\phi \wedge \phi_2}(r_2)) \tag{5.29}$$

$$\sigma_{\phi \wedge \phi_1}(r_1 \cap^{\otimes} r_2) = \sigma_{\phi}(\sigma_{\phi \wedge \phi_1}(r_1) \cap^{\otimes} \sigma_{\phi \wedge \phi_1}(r_2))\,. \tag{5.30}$$

Note that the preconditions in the above theorems are all necessary. Some equivalence results of classical relational algebra that generally do not carry over to our algebra on pcv-relations are the laws of *nested difference*, *idempotency*, and *de Morgan*. That is, for pcv-relations r_1, r_2, and r_3 of the same type R, in general *neither* of the following holds: $r_1 \cap^{\otimes} r_2 = r_1 -^{\ominus} (r_1 -^{\ominus} r_2)$, $r_1 \cap^{\otimes} r_1 = r_1$, $r_1 \cup^{\oplus} r_1 = r_1$, $r_1 -^{\ominus} (r_2 \cap^{\otimes} r_3) = (r_1 -^{\ominus} r_2) \cup^{\oplus} (r_1 -^{\ominus} r_3)$, and $r_1 -^{\ominus} (r_2 \cup^{\oplus} r_3) = (r_1 -^{\ominus} r_2) \cap^{\otimes} (r_1 -^{\ominus} r_3)$.

6 Query Optimization

We have seen that our algebraic operations on pcv-relations have similar computational costs as their classical counterparts in relational algebra. Moreover, it turned out that most of the query equivalences of classical relational algebra also hold for our algebra on pcv-relations. Hence, many query optimization techniques for classical relational algebra can easily be generalized to our algebra on pcv-relations.

Consider, for example, the heuristic rule to move in particular selections but also projections as much inside a given query expression as possible, in order to reduce the number of tuples and attributes, respectively, in subsequent operations. This can be done by the same query equivalences as in classical relational algebra:

- We can use (5.2) to break up conjunctive selection conditions. Note that the new selections produced this way are often easier to move inside the query expression.
- We can use (5.3), (5.5), (5.6), (5.10), (5.14), (5.20)–(5.22), and (5.25)–(5.30) to move selection operations as much inside the query expression as possible.
- We can use (5.9), (5.13), (5.17), and (5.19) to structure the query expression in a better way.
- We can use (5.4), (5.5), (5.7), (5.11), (5.15), and (5.23) to break down projections and to move projections as much inside the query expression as possible.

The following is an example of such a heuristic query optimization.

Example 6.1 Let $\mathbf{string}$ and $\mathbf{integer}$ be atomic types. Let date and $\mathsf{arrangement}$ be defined as follows:

$$\begin{aligned} \mathsf{date} &= [\mathsf{month}\colon \mathbf{integer}, \mathsf{day}\colon \mathbf{integer}, \mathsf{year}\colon \mathbf{integer}] \\ \mathsf{arrangement} &= [\mathsf{lab}\colon \mathbf{string}, \mathsf{date}\colon \mathsf{date}, \mathsf{assistants}\colon \{\mathbf{string}\}]\,. \end{aligned}$$

Let R_1, R_2, R_3, and S denote the following tuple types:

$$\begin{aligned} R_1 &= [\mathsf{substance\text{-}id}\colon \mathbf{string}, \alpha\text{-}\mathsf{arrangement}\colon \mathsf{arrangement}, \alpha\text{-}\mathsf{result}\colon \mathbf{string}] \\ R_2 &= [\mathsf{substance\text{-}id}\colon \mathbf{string}, \beta\text{-}\mathsf{arrangement}\colon \mathsf{arrangement}, \beta\text{-}\mathsf{result}\colon \mathbf{string}] \\ R_3 &= [\mathsf{substance\text{-}id}\colon \mathbf{string}, \gamma\text{-}\mathsf{arrangement}\colon \mathsf{arrangement}, \gamma\text{-}\mathsf{result}\colon \mathbf{string}] \\ S &= [\mathsf{substance\text{-}id}\colon \mathbf{string}, \alpha\text{-}\mathsf{result}\colon \mathbf{string}, \beta\text{-}\mathsf{result}\colon \mathbf{string}, \gamma\text{-}\mathsf{result}\colon \mathbf{string}]\,. \end{aligned}$$

Let r_1, r_2, and r_3 be pcv-relations over R_1, R_2, and R_3, respectively. Let ϕ_1 and ϕ_2 denote the selection conditions $x.\mathsf{data}.\alpha\text{-}\mathsf{result} = {'+'}$ and $x.\mathsf{data}.\beta\text{-}\mathsf{result} = x.\mathsf{data}.\gamma\text{-}\mathsf{result}$, respectively.

Then, the query expression

$$\sigma_{\phi_1 \wedge \phi_2}(\pi_S^{\oplus_{pc}}((r_1 \bowtie^{\otimes_{pc}} r_2) \bowtie^{\otimes_{pc}} r_3))$$

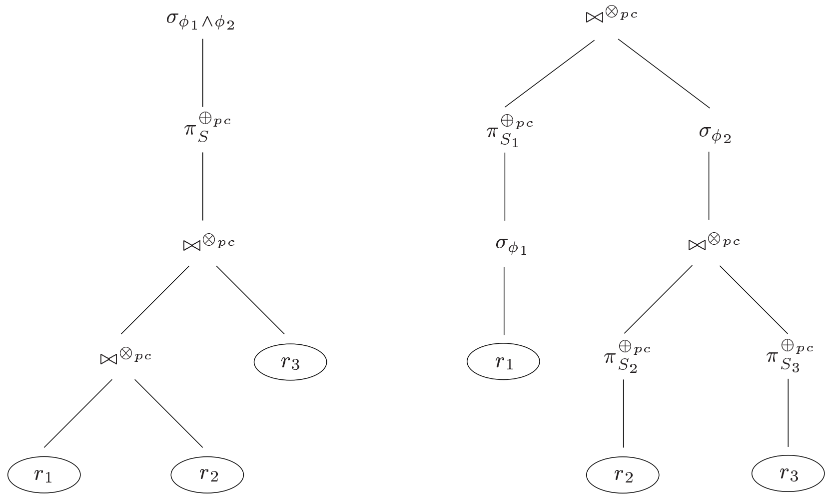

Figure 3: Two equivalent query trees

can be transformed into the following equivalent query expression by reorganizing the two joins and by pushing the selections and projections through them (see also Figure 3):

$$\pi_{S_1}^{\oplus_{pc}}(\sigma_{\phi_1}(r_1)) \bowtie^{\otimes_{pc}} \sigma_{\phi_2}(\pi_{S_2}^{\oplus_{pc}}(r_2) \bowtie^{\otimes_{pc}} \pi_{S_3}^{\oplus_{pc}}(r_3)),$$

where the tuple types S_1, S_2, and S_3 are given as follows:

$$\begin{aligned} S_1 &= [\mathsf{substance\text{-}id}\colon \mathbf{string},\ \alpha\text{-}\mathsf{result}\colon \mathbf{string}] \\ S_2 &= [\mathsf{substance\text{-}id}\colon \mathbf{string},\ \beta\text{-}\mathsf{result}\colon \mathbf{string}] \\ S_3 &= [\mathsf{substance\text{-}id}\colon \mathbf{string},\ \gamma\text{-}\mathsf{result}\colon \mathbf{string}]\,. \end{aligned}$$

Note that this transformation is based on the fact that the probabilistic conjunction strategy $\otimes_{pc}$ is distributive over the probabilistic disjunction strategy $\oplus_{pc}$.

7 Related Work

In this section, we give a brief comparison to related work in the literature.

7.1 ProbView

The work closest in spirit to this paper is perhaps the one on ProbView by Lakshmanan et al. [17], which we have already introduced in Section 1. There are, however, several important differences, which can be briefly summarized as follows:

- Our approach combines probabilistic uncertainty with complex value relations, while ProbView is only a probabilistic generalization of flat relations.

• Rather than defining probabilistic conjunction and disjunction strategies by a rigorous foundation on probability theory, [17] gives an axiomatic characterization of suitable such strategies.

• We discovered that there are combinations of probabilistic pairs and dependence information that are unsatisfiable. In [17], in contrast, this important fact is ignored.

• Differently from [17], we introduce probabilistic difference strategies that are not necessarily derived from probabilistic conjunction strategies and probabilistic negation.

• In certain cases, our approach yields tighter probability intervals than ProbView.

• We give a detailed picture of query equivalences that hold in our algebra, while [17] presents only a few number of equivalence results, which center around compaction.

Note that our approach can be seen as a generalization of ProbView in the sense that it allows complex values and more refined probabilistic combination strategies.

We now describe in more detail the crucial differences between the probabilistic background of our approach and the one of ProbView.

In [17], a probabilistic conjunction (resp., disjunction) strategy is defined as a function $\otimes$ (resp., $\oplus$) that assigns any two probabilistic pairs $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])$ a probabilistic pair $(e_1 \wedge e_2, [l, u])$ (resp., $(e_1 \vee e_2, [l, u])$) such that the postulates BOTTOMLINE, IGNORANCE, IDENTITY, ANNIHILATOR, COMMUTATIVITY, ASSOCIATIVITY, and MONOTONICITY shown in Table 10 hold for all probabilistic pairs $p_1 = (e_1, [l_1, u_1])$, $p_1^\star = (e_1, [l_1^\star, u_1^\star])$, $p_2 = (e_2, [l_2, u_2])$, and $p_3 = (e_3, [l_3, u_3])$.

Our rigorous foundation on probability theory now allows us to verify this axiomatic approach of ProbView. Indeed, by Proposition 2.12, every probabilistic conjunction and disjunction strategy in our framework satisfies the postulates IGNORANCE and ANNIHILATOR. However, rather than satisfying IDENTITY, our probabilistic conjunction and disjunction strategies satisfy only WEAK IDENTITY (see Example 2.13).

Moreover, there are probabilistic conjunction and disjunction strategies $\otimes$ and $\oplus$ in our framework that do *not* satisfy BOTTOMLINE, as the following example shows.

Example 7.1 We show that the probabilistic conjunction and disjunction strategies for left implication do not satisfy BOTTOMLINE. Let the two probabilistic pairs p_1 and p_2 be given by $(e_1, [.2, .7])$ and $(e_2, [.5, .8])$, respectively. It can easily be verified that $\{b \rightarrow a, (a, [.2, .7]), (b, [.5, .8])\}$ is satisfiable. Moreover, we get:

$$\begin{aligned} p_1 \otimes_{li} p_2 &= (e_1 \wedge e_2, [.5, .7]) > (e_1 \wedge e_2, [.2, .7]) = p_1 \otimes_{pc} p_2\,, \\ p_1 \oplus_{li} p_2 &= (e_1 \vee e_2, [.5, .7]) < (e_1 \vee e_2, [.5, .8]) = p_1 \oplus_{pc} p_2\,. \end{aligned}$$

The next example shows that some of our probabilistic conjunction and disjunction strategies also do *not* satisfy COMMUTATIVITY (that is, they are *not* commutative).

Example 7.2 We show that the static probabilistic conjunction and disjunction strategies for left implication do not satisfy COMMUTATIVITY. Let the two probabilistic pairs p_1 and p_2 be given by $(e_1, [.2, .7])$ and $(e_2, [.5, .6])$, respectively. It is now easy to see that $\{b \rightarrow a, (a, [.2, .7]), (b, [.5, .6])\}$ is satisfiable. Moreover, we get:

$$\begin{aligned} p_1 \otimes_{li} p_2 &= (e_1 \wedge e_2, [.5, .6]) \not\equiv (e_2 \wedge e_1, [.2, .6]) = p_2 \otimes_{li} p_1\,, \\ p_1 \oplus_{li} p_2 &= (e_1 \vee e_2, [.5, .7]) \not\equiv (e_2 \vee e_1, [.5, .6]) = p_2 \oplus_{li} p_1\,. \end{aligned}$$

Table 10: ProbView postulates for probabilistic combination strategies

BOTTOMLINE	$p_1 \otimes p_2 \;\leq\; p_1 \otimes_{pc} p_2$ $p_1 \oplus p_2 \;\geq\; p_1 \oplus_{pc} p_2$
IGNORANCE	$p_1 \otimes p_2 \;\subseteq\; p_1 \otimes_{ig} p_2$ $p_1 \oplus p_2 \;\subseteq\; p_1 \oplus_{ig} p_2$
IDENTITY	$p_1 \otimes p_2 \;\equiv\; p_1,$ if $[l_2, u_2] = [1, 1]$ and $e_1 \wedge e_2$ is consistent $p_1 \oplus p_2 \;\equiv\; p_1,$ if $[l_2, u_2] = [0, 0]$
ANNIHILATOR	$p_1 \otimes p_2 \;\equiv\; \bot,$ if $[l_2, u_2] = [0, 0]$ $p_1 \oplus p_2 \;\equiv\; \top,$ if $[l_2, u_2] = [1, 1]$
COMMUTATIVITY	$p_1 \otimes p_2 \equiv p_2 \otimes p_1$ $p_1 \oplus p_2 \equiv p_2 \oplus p_1$
ASSOCIATIVITY	$(p_1 \otimes p_2) \otimes p_3 \equiv p_1 \otimes (p_2 \otimes p_3)$ $(p_1 \oplus p_2) \oplus p_3 \equiv p_1 \oplus (p_2 \oplus p_3)$
MONOTONICITY	$p_1 \otimes p_2 \leq p_1^\star \otimes p_2,$ if $[l_1, u_1] \leq [l_1^\star, u_1^\star]$ $p_1 \oplus p_2 \leq p_1^\star \oplus p_2,$ if $[l_1, u_1] \leq [l_1^\star, u_1^\star]$

In summary, these observations show that ProbView assumes a number of postulates that do not have any justification in probability theory. Moreover, as ProbView is missing a rigorous probabilistic background, it also ignores the important fact that there are combinations of probabilistic pairs $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])$ with dependence information $KB \subseteq KB^\star(e_1, e_2)$ that are unsatisfiable.

Finally, we provide an example in which our rigorous approach based on probability theory yields tighter intervals than the axiomatic one of ProbView.

Example 7.3 Consider the case of left implication. Let the probabilistic pairs p_1 and p_2 be given by $(e_1, [.2, .7])$ and $(e_2,\, [.5, .6])$, respectively. It is easy to verify that $\{b \,\rightarrow\, a,\ (a, [.2, .7]),\ (b, [.5, .6])\}$ is satisfiable.

Under the static probabilistic conjunction, disjunction, and difference strategies for left implication, we obtain the intervals $[.5, .6]$, $[.5, .7]$, and $[0, .2]$, respectively. ProbView, in contrast, just produces $[.2, .6]$, $[.5, .7]$, and $[0, .5]$, respectively (see Table 11).

7.2 Other Related Work

As argued in [17], ProbView generalizes various other approaches, most notably the ones by Cavallo and Pittarelli [5] and by Barbará et al. [3]. In detail, Cavallo and Pittarelli [5] view relations in a (flat) relational database as probability distribution functions, where tuples in the same relation are viewed as pairwise disjoint events whose probabilities sum up to 1. Drawbacks of this approach have been pointed out in [6]. An extension of the model using probability intervals, which are viewed as constraints on the probabilities, is reviewed in [21]. Barbará et al. [3] have considered a probabilistic extension to the relational model, in

Table 11: Comparison with ProbView

Our Approach	ProbView
$p_1 \otimes_{li} p_2 \;=\; (e_1 \wedge e_2, [\mathbf{.5, .6}])$	$p_1 \otimes_{pc} p_2 \;=\; (e_1 \wedge e_2, [\mathbf{.2, .6}])$
$p_1 \oplus_{li} p_2 \;=\; (e_1 \vee e_2, [.5, .7])$	$p_1 \oplus_{pc} p_2 \;=\; (e_1 \vee e_2, [.5, .7])$
$p_1 \ominus_{li} p_2 \;=\; (e_1 \wedge \neg e_2, [\mathbf{0, .2}])$	$p_1 \ominus p_2 \;=\; (e_1, [.2, .7]) \otimes_{ig} (\neg e_2, [.4, .5]) \;=\; (e_1 \wedge \neg e_2, [\mathbf{0, .5}])$

which imprecise attributes are modeled as probability distributions over finite sets of values. Their approach assumes that key attributes are deterministic (have probability 1) and that non-key attributes in different relations are independent. No probabilities can be assigned to outmost tuples.

Fuhr and Rölleke [10] consider a probabilistic version of NF2 relations, extending their approach for flat tuples [11], and define a relational algebra for this model. Probabilities are assigned to tuples and to values of nested tuples (that is, set-valued attributes), which are viewed as events that have an associated event expression. The algebraic operators manipulate tuples by combining value and event expressions appropriately. The probability of a derived tuple is computed from the probabilities of initial tuples by taking its event expression into consideration. The approach in [10] defines an intensional semantics in which probabilities are defined through possible worlds. The evaluation method assumes that in nondeterministic relations (that is, relations with uncertain tuples), joint occurrence of two different values is either always independent or impossible. Our approach has no such severe restriction (we do not make any independence or mutual exclusion assumptions). Moreover, [10] does not use probability intervals but a single probability value. On the other hand, the algebra in [10] provides nest and unnest operators, which we have not done here.

We see as major drawbacks of the approach in [10] the above non-dependency assumption on relations, which is rarely met in practice, and that the evaluation of event expressions takes exponential time in general, owing to complicated probability sums to be computed—this seems the price to pay for a smooth intensional semantics, though.

The probabilistic database model of Dey and Sarkar [6] assigns each tuple in a (flat) relational database a probability value in a special attribute. The classical relational operations are defined adopting different assumptions on the relationship between tuples; in particular, join assumes independence; union and difference assume positive correlation; and compaction assumes disjointness or positive correlation. Our model is more general, since it has complex values, intervals, and allows for using different strategies (reflecting different relationships between tuples) in the algebraic operations. Based on [6], a probabilistic extension to SQL is developed in [7].

Zimányi [25] presents an approach to querying probabilistic databases based on probabilistic first-order logic. A probabilistic database is axiomatized as a first-order probabilistic theory, which has a possible worlds semantics as in [14]. Tuples t in relations have a trace formula ϕ attached, which intuitively selects the worlds in which t is true. The classical relational algebra operations are defined using formulas and manipulate trace formulas. A query is evaluated in two steps: the first step yields a relation of tuples with trace formulas, which are then evaluated using the assertions of the probabilistic theory to obtain a probability for each tuple. Compared to our work, [25] aims at flat databases and considers no probability intervals (but mentions a possible extension). It assumes complete independence of occurrences of distinct tuples in the same or different relations. Moreover, as with most other approaches, no query equivalences are discussed.

8 Discussion and Conclusion

8.1 Summary

In this paper, we presented a probabilistic data model for complex values, which combines the concept of probabilistic relations in [17] with the idea of complex values [1]. We then introduced an algebra for querying database instances, which uses probabilistic combination strategies that are based on a rigorous probabilistic background. We showed that our data model and algebra for probabilistic complex values is a generalization of the classical relational data model and algebra. Moreover, we showed that under static combination strategies, all the algebraic operations can be polynomially computed with similar efficiency as in classical relational algebra. Finally, we presented query equivalences and discussed their use for query optimization. It turned out that most of the equivalences of classical relational algebra hold in our generalized model. Hence, many classical query optimization techniques carry over to our model.

8.2 Possible Generalizations

The concept of probabilistic combination strategies in our algebra is based on the idea that the computation of probability values can be reduced to a number of *local* computation steps, each of which is restricted to only two probabilistic pairs and some dependence information about them. An alternative way of computing probability values could be to solve a number of *global* optimization problems, subject to *all* the available probabilistic knowledge and dependence informations about the pcv-tuples in the database. In the following, we give a rough sketch of how this could be done.

Assume that we are given a pcv-database $d = \{r_1, \ldots, r_n\}$ and a query expression q over the pcv-relations $r_1, \ldots, r_n$.

We now construct a knowledge base KB that collects all probabilistic pairs in d and all dependence informations that are encoded in the combination strategies in q. We start by initializing KB with the set of all probabilistic pairs in d:

$$KB \ := \bigcup_{i \in \{1,\ldots,n\}} \{t.\mathsf{prob} \mid t \in r_i\}\,.$$

We next have to collect all the dependence informations that emerge from q. Unfortunately, some of them may depend on the contents of the intermediate relations in the evaluation of q on $r_1, \ldots, r_n$, and thus on probability values that are actually still to be computed. We thus evaluate q on *logical approximations* $r'_1, \ldots, r'_n$ of $r_1, \ldots, r_n$, which are obtained from $r_1, \ldots, r_n$ by setting each probability value in r_i to 1. Let (δ', μ') denote the result of evaluating q on $r'_1, \ldots, r'_n$.

For each application of a combination strategy, we then add some dependence information to KB as follows:

- For each evaluation of an expression of the kind $(e_1, [l_1, u_1]) \odot (e_2, [l_2, u_2])$, where $\odot$ is a probabilistic combination strategy for some $KB(a, b) \subseteq KB^\star(a, b)$, we perform

$$KB \ := \ KB \cup KB(e_1, e_2)\,.$$

- For each evaluation of an expression of the form $\bigoplus_{i\in I}(e_i, [l_i, u_i])$, where $\oplus$ is the probabilistic disjunction strategy for some $KB(a,b) \subseteq KB^\star(a,b)$, we perform

$$KB \ := \ KB \cup \bigcup\{KB(\bigvee_{i\in I_1} e_i, \bigvee_{i\in I_2} e_i) \mid \emptyset \neq I_1, I_2 \subseteq I,\ I_1 \cap I_2 = \emptyset\}\,.$$

Let for $p = (e, [1,1])$ denote $tlc(KB, p)$ the probabilistic pair $(e, [l,u])$ such that $KB \models_{\mathit{tight}} (e, [l,u])$. We are now ready to define the result $r = (\delta, \mu)$ of evaluating q on $r_1, \ldots, r_n$ as follows:

$$\delta \ = \ \{v \in \delta' \mid tlc(KB, \mu'(v)) \not\equiv \bot\},$$

$$\mu(v) \ = \ tlc(KB, \mu'(v)) \text{ for all } v \in \delta.$$

That is, for each pcv-tuple $t \in r$, two global (and, in general, nonlinear) optimization problems must be solved to compute the tight probability bounds $t.\mathsf{lb}$ and $t.\mathsf{ub}$.

The described technique is based on the idea that every complex value in the result of evaluating q on $r_1, \ldots, r_n$ should also be contained in the result of evaluating q on $r'_1, \ldots, r'_n$. Hence, it requires the following restrictions on the query expression q:

- We must assume that every selection condition in q is probability-free and that q does not contain any difference operations. Otherwise, some selection or difference operations in the evaluation of q on $r'_1, \ldots, r'_n$ may eliminate some complex values that should be represented in the result of evaluating q on $r_1, \ldots, r_n$.

- We must assume that every probabilistic combination strategy in q is of one of the cases ignorance, independence, positive correlation, and negative correlation, since otherwise the evaluation of q on $r'_1, \ldots, r'_n$ may run into unsatisfiability. Furthermore, too much dependence knowledge (i.e., more than actually expressed by q) may be generated.

We remark that for the evaluation of q on $r'_1, \ldots, r'_n$, the expression q can be transformed by using the same query equivalences as in classical relational algebra.

In summary, such a global approach can be defined only for a restricted sublanguage of our algebra on pcv-relations. Moreover, it produces in general quite large nonlinear optimization problems. Whether they can be solved efficiently, especially for large pcv-databases (e.g., ones containing more than a million pcv-tuples), remains doubtful. The global approach, however, generally produces tighter probability intervals than the local evaluation of our algebra on pcv-relations.

Hence, the use of the global approach seems to be limited to specialized knowledge representation tasks that involve little probabilistic knowledge and in which the probabilistic precision of the computed probability values is an important issue.

8.3 Future Work

Several issues remain for further investigation. One such issue are other operators besides those of classical relation algebra, such as NF2 nest and unnest or the operators in [10, 6, 3, 21]. Another issue is to explore further query optimization techniques. Moreover, it would be very interesting to more deeply investigate the relationship of our model and algebra to purely extensional and purely intensional approaches to probabilistic databases. Finally, an intriguing issue is to extend our approach to object-oriented databases (see [8] for preliminary work in this direction). To our knowledge, few approaches to probabilistic object-oriented databases have been proposed so far (see [15]).

A Appendix: Proofs for Section 2

Proof of Proposition 2.1. We first prove (2.1). Recall that $Pr \models PC(a,b)$ holds if and only if $Pr(a \wedge b) = \min(Pr(a), Pr(b))$. The latter is equivalent to $Pr(a \wedge b) = Pr(a)$ or $Pr(a \wedge b) = Pr(b)$, which in turn is equivalent to $Pr \models a \rightarrow b$ or $Pr \models b \rightarrow a$. The statement (2.2) can be proved similarly.

To prove (2.3), it remains by (2.1) to prove the statement for $F \in \{a \rightarrow b,\ b \rightarrow a\}$. Recall that $Pr \models a \rightarrow b$ iff $Pr(a \wedge b) = Pr(a)$ and $Pr \models Ind(a,b)$ iff $Pr(a \wedge b) = Pr(a) \cdot Pr(b)$.

Suppose first $Pr \models a \rightarrow b$. Then, $Pr(a) \cdot Pr(b) = Pr(a \wedge b)$ is equivalent to $Pr(b) = 1$ or $Pr(a \wedge b) = 0$. Since $Pr(a \wedge \neg b) = 0$, the latter is equivalent to $Pr(a \wedge \neg b) = \min(Pr(a), Pr(\neg b))$. Thus, $Pr \models Ind(a,b)$ iff $Pr \models NC(a,b)$.

Suppose next $Pr \models b \rightarrow a$. Then, $Pr(a) \cdot Pr(b) = Pr(a \wedge b)$ is equivalent to $Pr(a) = 1$ or $Pr(a \wedge b) = 0$. Since $Pr(\neg a \wedge b) = 0$, the latter is equivalent to $Pr(a \wedge \neg b) = \min(Pr(a), Pr(\neg b))$. Again, it follows that $Pr \models Ind(a,b)$ iff $Pr \models NC(a,b)$. The proof of (2.4) is similar. $\square$

Proof of Theorem 2.2. The statement is trivial for every $KB \subseteq \{a \rightarrow b,\ b \rightarrow a,\ a \wedge b \rightarrow \bot,\ \top \rightarrow a \vee b\}$ (set $n = 1$ and $KB_1 = KB$).

If $Ind(a,b) \in KB$, then by (2.3) and (2.4) it can be replaced in KB by either $NC(a,b)$ or $PC(a,b)$. If $PC(a,b) \in KB$, then by (2.1), KB is equivalent to the disjunction of $KB_1 = KB - \{PC(a,b)\} \cup \{a \rightarrow b\}$ and $KB_2 = KB - \{PC(a,b)\} \cup \{b \rightarrow a\}$. By (2.2), we can similarly express $NC(a,b) \in KB$ through disjunction. Recursively applied, we can eliminate $Ind(a,b)$, $PC(a,b)$, and $NC(a,b)$ from KB and obtain a disjunction of at most four dependence informations $KB_1, \ldots, KB_4$. $\square$

Proof of Lemma 2.3. For $KB = \{Ind(a,b)\}$, the statement is obviously true. Otherwise, by Theorem 2.2, the dependence information KB is equivalent to a logical disjunction of $KB_1, \ldots, KB_n \subseteq \{a \rightarrow b,\ b \rightarrow a,\ a \wedge b \rightarrow \bot,\ \top \rightarrow a \vee b\}$, where $1 \leq n \leq 4$. Hence, KB is satisfiable iff some KB_i with $1 \leq n \leq 4$ is satisfiable, which in turn is equivalent to some KB_i with $1 \leq n \leq 4$ being a proper subset of $\{a \rightarrow b,\ b \rightarrow a,\ a \wedge b \rightarrow \bot,\ \top \rightarrow a \vee b\}$. By the construction of the KB_i in the proof of Theorem 2.2, the latter is now equivalent to $\{a \rightarrow b,\ b \rightarrow a,\ a \wedge b \rightarrow \bot,\ \top \rightarrow a \vee b\} \not\subseteq KB$. $\square$

Proof of Proposition 2.4 (sketch). We show only (a) and (b). The other statements can be proved similarly.

(a) Assume first $KB \in \{\emptyset, \{Ind(a,b)\}\}$. A probabilistic interpretation Pr_1 that satisfies $KB \cup \{p_1, p_2\}$ is given as follows:

$$a \wedge b,\ a \wedge \neg b,\ \neg a \wedge b,\ \neg a \wedge \neg b \ \mapsto\ l_1 \cdot l_2,\ l_1 \cdot (1-l_2),\ (1-l_1) \cdot l_2,\ (1-l_1) \cdot (1-l_2)\,.$$

Suppose next $KB = \{PC(a,b)\}$. We have $l_1 \leq u_2$ or $l_2 \leq u_1$. Without loss of generality, let $l_1 \leq u_2$. A probabilistic interpretation Pr_2 that satisfies $KB \cup \{p_1, p_2\}$ is given as follows:

$$a \wedge b,\ a \wedge \neg b,\ \neg a \wedge b,\ \neg a \wedge \neg b \ \mapsto\ l_1,\ 0,\ u_2 - l_1,\ 1 - u_2\,.$$

Finally, the proof for the case $KB = \{NC(a,b)\}$ is analogous.

(b) Let $KB = \{a \rightarrow b\}$. Assume first that $l_1 \leq u_2$. The interpretation Pr_2 in the proof of (a) is clearly a model of $KB \cup \{p_1, p_2\}$. Conversely, assume that there is a model Pr of $KB \cup \{p_1, p_2\}$. It thus follows $l_1 \leq Pr(a) \leq Pr(b) \leq u_2$. Hence, we get $l_1 \leq u_2$. $\square$

Proof of Proposition 2.9 (sketch). We just derive the static probabilistic combination strategies for $KB = \{a \rightarrow b\}$. The derivations for the other cases are similar. Let $q_1 = (a, [l_1, u_1])$ and $q_2 = (b, [l_2, u_2])$.

We first show that the minimum (resp., maximum) of $Pr(a \wedge b)$ subject to all models Pr of $KB \cup \{q_1, q_2\}$ is given by l_1 (resp., $\min(u_1, u_2)$). Let Pr be a model of $KB \cup \{q_1, q_2\}$. Then, it follows that

$$l_1 \leq Pr(a) = Pr(a \wedge b) \leq \min(Pr(a), Pr(b)) \leq \min(u_1, u_2) .$$

These bounds are tight. For example, a probabilistic interpretation Pr_1 such that $Pr_1(a \wedge b) = l_1$ and $Pr_1 \models KB \cup \{q_1, q_2\}$ is given by:

$$a \wedge b,\ a \wedge \neg b,\ \neg a \wedge b,\ \neg a \wedge \neg b \ \mapsto\ l_1,\ 0,\ u_2 - l_1,\ 1 - u_2 .$$

Moreover, a probabilistic interpretation Pr_2 such that $Pr_2(a \wedge b) = \min(u_1, u_2)$ and $Pr_2 \models KB \cup \{q_1, q_2\}$ is given by:

$$a \wedge b,\ a \wedge \neg b,\ \neg a \wedge b,\ \neg a \wedge \neg b \ \mapsto\ \begin{cases} u_1,\ 0,\ u_2 - u_1,\ 1 - u_2, & \text{if } u_1 \leq u_2; \\ u_2,\ 0,\ 0,\ 1 - u_2, & \text{otherwise.} \end{cases}$$

We next show that the minimum (resp., maximum) of $Pr(a \vee b)$ subject to all models Pr of $KB \cup \{q_1, q_2\}$ is given by $\max(l_1, l_2)$ (resp., u_2). Let Pr be a model of $KB \cup \{q_1, q_2\}$. It then follows:

$$\max(l_1, l_2) \leq \max(Pr(a), Pr(b)) \leq Pr(a \vee b) = Pr(b) \leq u_2 .$$

The probabilistic interpretation Pr_1 as defined above satisfies $Pr_1(a \vee b) = u_2$ and $Pr_1 \models KB \cup \{q_1, q_2\}$. A probabilistic interpretation Pr_3 such that $Pr_3(a \vee b) = \max(l_1, l_2)$ and $Pr_3 \models KB \cup \{q_1, q_2\}$ is given by:

$$a \wedge b,\ a \wedge \neg b,\ \neg a \wedge b,\ \neg a \wedge \neg b \ \mapsto\ \begin{cases} l_1,\ 0,\ l_2 - l_1,\ 1 - l_2, & \text{if } l_1 \leq l_2; \\ l_1,\ 0,\ 0,\ 1 - l_1, & \text{otherwise.} \end{cases}$$

Finally, the minimum and maximum of $Pr(a \wedge \neg b)$ subject to all models Pr of $KB \cup \{q_1, q_2\}$ are both given by 0, since $KB \cup \{q_1, q_2\}$ is always satisfiable. □

Proof of Proposition 2.10 (sketch). The statements can be easily verified along the static probabilistic combination strategies shown in Table 1 (see Proposition 2.9). □

Proof of Proposition 2.11. Let $KB(e_1, e_2)$ denote any set among $\emptyset$, $\{Ind(e_1, e_2)\}$, $\{PC(e_1, e_2)\}$, $\{e_2 \rightarrow e_1, e_1 \rightarrow e_2\}$, $\{NC(e_1, e_2)\}$, $\{e_1 \wedge e_2 \rightarrow \bot\}$, $\{\top \rightarrow e_1 \vee e_2\}$, and $\{e_1 \wedge e_2 \rightarrow \bot, \top \rightarrow e_1 \vee e_2\}$. It is now easy to verify that a probabilistic interpretation Pr satisfies $KB(e_1, e_2) \cup \{(e_1, [l_1, u_1]), (e_2, [l_2, u_2])\}$ iff it satisfies $KB(e_2, e_1) \cup \{(e_2, [l_2, u_2]), (e_1, [l_1, u_1])\}$. Hence, the minimum (resp., maximum) of $Pr(e_1 \wedge e_2)$ and $Pr(e_1 \vee e_2)$ subject to all models Pr of $KB(e_1, e_2) \cup \{(e_1, [l_1, u_1]), (e_2, [l_2, u_2])\}$ coincides with the minimum (resp., maximum) of $Pr(e_2 \wedge e_1)$ and $Pr(e_2 \vee e_1)$ subject to all models Pr of $KB(e_2, e_1) \cup \{(e_2, [l_2, u_2]), (e_1, [l_1, u_1])\}$. □

Proof of Proposition 2.12 (sketch). We just prove the properties listed at first place under IGNORANCE, WEAK IDENTITY, and ANNIHILATOR. The proof of the other properties is similar. In the sequel, let $q_1 = (a, [l_1, u_1])$ and $q_2 = (b, [l_2, u_2])$.

IGNORANCE: The tight interval for $Pr(a \wedge b)$ over all models Pr of $KB \cup \{q_1, q_2\}$ is contained in the tight interval for $Pr(a \wedge b)$ over all models Pr of $\{q_1, q_2\}$. This shows $p_1 \otimes p_2 \subseteq p_1 \otimes_{ig} p_2$.

IDENTITY: Since $[l_1, u_1] = [1, 1]$, the tight interval for $Pr(a \wedge b)$ subject to all models Pr of $KB \cup \{q_1, q_2\}$ is contained in $[l_2, u_2]$. Assume now that $a \wedge b \rightarrow \perp \notin KB$ and $a \rightarrow b \notin KB$. Let $r \in \{l_2, u_2\}$. A probabilistic interpretation Pr_1 such that $Pr_1(a \wedge b) = r$ and $Pr_1 \models KB \cup \{q_1, q_2\}$ is given as follows:

$$a \wedge b,\ a \wedge \neg b,\ \neg a \wedge b,\ \neg a \wedge \neg b \ \mapsto\ r,\ 1 - r,\ 0,\ 0\,.$$

In summary, it follows $p_1 \otimes p_2 \ \equiv\ p_2$.

ANNIHILATOR: Since $[l_1, u_1] = [0, 0]$ or $[l_2, u_2] = [0, 0]$, the tight interval for $Pr(a \wedge b)$ over all models Pr of $KB \cup \{q_1, q_2\}$ is given by $[0, 0]$. This shows $p_1 \otimes p_2 \ \equiv\ \perp$. ☐

B Appendix: Proofs for Section 4

Proof of Lemma 4.13 (sketch). The statement can easily be verified along the static probabilistic combination strategies for ignorance, independence, positive correlation, and negative correlation shown in Table 1 (see Proposition 2.9). ☐

Proof of Theorem 4.14. Equations (4.1) and (4.3) trivially hold. Moreover, Equations (4.2) and (4.4)–(4.8) follow immediately from Lemma 4.13. We remark in particular that $\varepsilon(\phi)$ is a selection condition that is applicable to $\varepsilon(R_1)$, that $\varepsilon(T)$ is a subtype of $\varepsilon(R_1)$, that N is a renaming condition for $\varepsilon(R_1)$, and that R_1 and R_2 are join-compatible in (4.4) and Cartesian-product-compatible in (4.5). ☐

Proof of Theorem 4.15. In the sequel, let $q_1 = (a, [l_1, u_1])$ and $q_2 = (b, [l_2, u_2])$.

(a) By Proposition 2.4 (a), the claim trivially holds for $KB = \{Ind(a, b)\}$ (in fact, this is true for every KB there).

Assume now $KB \neq \{Ind(a, b)\}$. By Theorem 2.2, there are sets $KB_1, \ldots, KB_n \subseteq \{a \rightarrow b,\ b \rightarrow a,\ a \wedge b \rightarrow \perp,\ \top \rightarrow a \vee b\}$, where $1 \leq n \leq 4$, such that for every probabilistic interpretation Pr it holds that $Pr \models KB$ iff $Pr \models KB_1$ or $\cdots$ or $Pr \models KB_n$. Hence, $KB \cup \{q_1, q_2\}$ is satisfiable iff $KB_i \cup \{q_1, q_2\}$ is satisfiable for some $i \in \{1, \ldots, n\}$.

We now show that for any given $KB' \subseteq \{a \rightarrow b,\ b \rightarrow a,\ a \wedge b \rightarrow \perp,\ \top \rightarrow a \vee b\}$, deciding whether $KB' \cup \{q_1, q_2\}$ is satisfiable can be done in time $O(1)$.

For $|KB'| \leq 1$, the claim is immediate by Proposition 2.4 (a)–(c), (e), and (f). Moreover, for $|KB'| = 4$, the set $KB' \cup \{q_1, q_2\}$ is unsatisfiable (cf. Lemma 2.3).

Assume next that $|KB'| = 2$. For $KB' = \{a \rightarrow b, b \rightarrow a\}$ and $KB' = \{a \wedge b \rightarrow \perp, \top \rightarrow a \vee b\}$, the claim is immediate by Proposition 2.4 (d) and (g). Any other KB' with $|KB'| = 2$ is logically equivalent to a set of formulas among $\{\top \rightarrow a\}$, $\{\top \rightarrow b\}$, $\{a \rightarrow \perp\}$, and $\{b \rightarrow \perp\}$. Then, $KB' \cup \{q_1, q_2\}$ is satisfiable iff $u_1 = 1$, $u_2 = 1$, $l_1 = 0$, and $l_2 = 0$, respectively.

Assume finally that $|KB'| = 3$. That is, KB' is logically equivalent to a set among $\{\top \rightarrow a \wedge b\}$, $\{\top \rightarrow a \wedge \neg b\}$, $\{\top \rightarrow \neg a \wedge b\}$, and $\{\top \rightarrow \neg a \wedge \neg b\}$. It is now easy to verify that then $KB' \cup \{q_1, q_2\}$ is satisfiable iff $u_1 \cdot u_2 = 1$, $u_1 \cdot (1 - l_2) = 1$, $(1 - l_1) \cdot u_2 = 1$, and $(1 - l_1) \cdot (1 - l_2) = 1$, respectively.

Thus, deciding whether each $KB_i \cup \{q_1, q_2\}$ is satisfiable is possible in time $O(1)$. In summary, deciding whether $KB \cup \{q_1, q_2\}$ is satisfiable is possible in time $O(1)$.

(b) We prove the claim only for probabilistic conjunction strategies. The proofs for probabilistic disjunction and difference strategies are similar.

The claim is immediate for $KB = \{Ind(a, b)\}$, since the static probabilistic combination strategies for independence are clearly computable in time $O(1)$ (see Table 1; in fact, this holds for every KB there).

Assume now $KB \neq \{Ind(a, b)\}$. By Theorem 2.2, there are sets $KB_1, \ldots, KB_n \subseteq \{a \to b, b \to a, a \wedge b \to \bot, \top \to a \vee b\}$, where $1 \leq n \leq 4$, such that for every probabilistic interpretation Pr it holds that $Pr \models KB$ iff $Pr \models KB_1$ or $\cdots$ or $Pr \models KB_n$. For every $i \in \{1, \ldots, n\}$, let $l_i^\star$ and $u_i^\star$ be defined by $KB_i \cup \{q_1, q_2\} \models_{\mathit{tight}} (a \wedge b, [l_i^\star, u_i^\star])$. That is, $l_i^\star$ and $u_i^\star$ are given by 1 and 0, respectively, if $KB_i \cup \{q_1, q_2\}$ is unsatisfiable, and they are defined by $q_1 \otimes_{KB_i} q_2 = (a \wedge b, [l_i^\star, u_i^\star])$, otherwise. It then follows:

$$q_1 \otimes_{KB} q_2 \;=\; (a \wedge b, [\min(l_1^\star, \ldots, l_n^\star), \max(u_1^\star, \ldots, u_n^\star)])\,.$$

That is, $q_1 \otimes_{KB} q_2$ can be computed as follows. For every $i \in \{1, \ldots, n\}$, we check whether $KB_i \cup \{q_1, q_2\}$ is satisfiable. If this is the case, we also compute $q_1 \otimes_{KB_i} q_2 = (a \wedge b, [l_i^\star, u_i^\star])$. Finally, we compute $\min(l_1^\star, \ldots, l_n^\star)$ and $\max(u_1^\star, \ldots, u_n^\star)$. By (a), deciding whether $KB_i \cup \{q_1, q_2\}$ is satisfiable can be done in time $O(1)$.

We now show that for any given $KB' \subseteq \{a \to b, b \to a, a \wedge b \to \bot, \top \to a \vee b\}$ such that $KB' \cup \{q_1, q_2\}$ is satisfiable, computing $q_1 \otimes_{KB'} q_2$ is possible in time $O(1)$.

Clearly, the static probabilistic combination strategies for ignorance, left implication, right implication, mutual exclusion, exhaustion, equivalence, and antivalence are all computable in constant time (see Table 1).

Assume next that KB' is logically equivalent to a set among $\{\top \to a\}$, $\{\top \to b\}$, $\{a \to \bot\}$, and $\{b \to \bot\}$. In these cases, we get $q_1 \otimes_{KB'} q_2 = (a \wedge b, [l, u])$, where $[l, u]$ is given by $[l_2, u_2]$, $[l_1, u_1]$, $[0, 0]$, and $[0, 0]$, respectively.

Suppose finally that KB' is logically equivalent to a set among the sets $\{\top \to a \wedge b\}$, $\{\top \to a \wedge \neg b\}$, $\{\top \to \neg a \wedge b\}$, and $\{\top \to \neg a \wedge \neg b\}$. Then, we get $q_1 \otimes_{KB'} q_2 = (a \wedge b, [l, u])$, where $[l, u]$ is given by $[1, 1]$, $[0, 0]$, $[0, 0]$, and $[0, 0]$, respectively.

Hence, computing each $q_1 \otimes_{KB_i} q_2$, where $KB_i \cup \{q_1, q_2\}$ is satisfiable, can be done in time $O(1)$. In summary, computing $q_1 \otimes_{KB} q_2$ is possible in time $O(1)$. □

Proof of Theorem 4.16. (a) The compaction $\kappa^{\oplus}(r)$ can be computed by partitioning r into sets of pcv-tuples with the same data value, merging the pcv-tuples in each set by applying the disjunction strategy $\oplus$, and removing all pcv-tuples t with $t.\mathsf{prob} \equiv \bot$. By Theorem 4.15, the disjunction strategy $\oplus$ can be applied in constant time. The other operations can clearly be done in polynomial time. Hence, in summary, the compaction $\kappa^{\oplus}(r)$ can be computed in polynomial time.

(b) The selection $\sigma_\phi(r)$ can be computed by first evaluating ϕ on every $t \in r$, and then removing all t from r with $t \not\models \phi$. This can be done in linear time if ϕ is fixed and sets of complex values are represented as totally ordered sets or as bitvectors.

(c) Any renaming operation $\rho_N(r)$ with N can be done in linear time: The set of top-level attributes $\{A_1, \ldots, A_k\}$ of the type of r which have to be changed according to N can be determined in linear time in the size of N. For each pcv-tuple t, the replacement of $\{A_1, \ldots, A_k\}$ by the new attribute names takes k steps, thus time linear in the size of t. Overall, the renaming $\rho_N(r)$ can be done in time linear in the size of r and N.

(d) The projection $\pi_S^{\oplus}(r)$ can be computed by first replacing $t.\mathsf{data}$ in each pcv-tuple $t \in r$ by $\pi_S(t.\mathsf{data})$, and then performing a compaction under $\oplus$. The former can clearly be done in polynomial time. By (a), the latter can also be done in polynomial time. □

Proof of Theorem 4.17. (a) As Cartesian product is a special case of join, it is sufficient to show the claim for join. The join $r_1 \bowtie^{\otimes} r_2$ can be computed as follows. For each $t_1 \in r_1$ and $t_2 \in r_2$, we check whether (i)

$t_1.\mathsf{data}.A = t_2.\mathsf{data}.A$ for all common top-level attribute names A, and (ii) $t_1.\mathsf{prob} \otimes t_2.\mathsf{prob} \not\equiv \bot$. If this is the case, then the pcv-tuple t that is defined by $t.\mathsf{data} = t_1.\mathsf{data} \bowtie t_2.\mathsf{data}$ and $t.\mathsf{prob} = t_1.\mathsf{prob} \otimes t_2.\mathsf{prob}$ belongs to the result. By Theorem 4.15, the conjunction strategy $\otimes$ can be applied in constant time. The other operations can clearly be done in polynomial time. Hence, in summary, the join $r_1 \bowtie^{\otimes} r_2$ can be computed in polynomial time (in fact, within the same order of magnitude as join in classical relational algebra).

(b) We show the claim for intersection. The proofs for union and difference are similar. The intersection $r_1 \cap^{\otimes} r_2$ can be computed as follows. For every $t_1 \in r_1$ for which there exists some $t_2 \in r_2$ such that $t_1.\mathsf{data} = t_2.\mathsf{data}$ and $t_1.\mathsf{prob} \otimes t_2.\mathsf{prob} \not\equiv \bot$, we include the pcv-tuple t that is defined by $t.\mathsf{data} = t_1.\mathsf{data}$ and $t.\mathsf{prob} = t_1.\mathsf{prob} \otimes t_2.\mathsf{prob}$ in the result. By Theorem 4.15, the conjunction strategy $\otimes$ can be applied in constant time. The other operations can clearly be done in polynomial time. Hence, in summary, the intersection $r_1 \cap^{\otimes} r_2$ can be computed in polynomial time (in fact, within the same order of magnitude as intersection in classical relational algebra). $\Box$

Proof of Theorem 4.18. We first show membership in NP. The set $KB(e_1, e_2) \cup \{p_1, p_2\}$ is satisfiable iff the following two conditions hold:

- there is a set F of satisfiable members of $\{e_1 \wedge \neg e_2,\ \neg e_1 \wedge e_2,\ e_1 \wedge e_2,\ \neg e_1 \wedge \neg e_2\}$,
- $KB'(a, b) \cup \{(a, [l_1, u_1]), (b, [l_2, u_2])\}$ is satisfiable, where $KB'(a, b) \subseteq KB^\star(a, b)$ is obtained from $KB(a, b)$ by adding $a \to b$, $b \to a$, $a \wedge b \to \bot$, and $\top \to a \vee b$ iff $e_1 \wedge \neg e_2$, $\neg e_1 \wedge e_2$, $e_1 \wedge e_2$, and $\neg e_1 \wedge \neg e_2$, respectively, are not contained in F.

Hence, membership in NP follows immediately from Theorem 4.15, since F can clearly be guessed and verified by a nondeterministic algorithm in polynomial time.

To show NP-hardness, we give a polynomial transformation from the NP-complete problem of deciding whether a propositional formula in CNF is satisfiable [13].

Let ϕ be a propositional formula in CNF over $\mathcal{B}$, and let $KB(a, b) \subseteq KB^\star(a, b)$ be any fixed satisfiable dependence information on two distinct basic events a and b. Consider first $KB(a, b) = \{Ind(a, b)\}$. Let $p_1 = p_2 = (\phi, [1, 1])$ and $e_1 = e_2 = \phi$.

We now show that ϕ is satisfiable iff $KB(e_1, e_2) \cup \{p_1, p_2\}$ is satisfiable. Assume first that ϕ is satisfiable. That is, there is a classical interpretation J such that $J \models \phi$. Hence, the probabilistic interpretation Pr defined by $Pr(J) = 1$, and $Pr(I) = 0$ for all $I \in \mathcal{I}_{\mathcal{B}}$ with $I \neq J$ is a model of $KB(e_1, e_2) \cup \{p_1, p_2\}$. Conversely, suppose that $KB(e_1, e_2) \cup \{p_1, p_2\}$ has a model Pr. Hence, in particular $Pr(\phi) = 1$, and thus there is a classical interpretation I such that $I \models \phi$ and $Pr(I) > 0$. That is, ϕ is satisfiable.

Consider now $KB(a, b) \neq \{Ind(a, b)\}$. Let KB_1 be the dependency information from the logical disjunction in Theorem 2.2 that is equivalent to $KB(a, b)$. By Lemma 2.3, we may assume without loss of generality that KB_1 is a proper subset of $\{a \to b,\ b \to a,\ a \wedge b \to \bot,\ \top \to a \vee b\}$. Define now p_1 and p_2 as follows:

$$p_1, p_2 = \begin{cases} (\phi, [1,1]), (\phi, [1,1]) & \text{if } a \wedge b \to \bot \notin KB_1; \\ (\neg\phi, [0,0]), (\neg\phi, [0,0]) & \text{if } a \wedge b \to \bot \in KB_1 \text{ and } \top \to a \vee b \notin KB_1; \\ (\phi, [1,1]), (\neg\phi, [0,0]) & \text{if } a \wedge b \to \bot,\ \top \to a \vee b \in KB_1 \text{ and} \\ & \quad a \to b \notin KB_1; \\ (\neg\phi, [0,0]), (\phi, [1,1]) & \text{otherwise.} \end{cases}$$

Then, by a similar line of argumentation as above, we obtain that ϕ is satisfiable iff $KB(e_1, e_2) \cup \{p_1, p_2\}$ is satisfiable, where e_1 and e_2 are the events of p_1 and p_2, respectively. Clearly, p_1 and p_2 are constructible in polynomial time from ϕ. This proves the hardness part under the asserted restriction. $\square$

Proof of Theorem 4.19. Let the dependence information $KB'(a, b) \subseteq KB^\star(a, b)$ be obtained from $KB(a, b)$ by adding the dependence formulas $a \to b$, $b \to a$, $a \wedge b \to \bot$, and $\top \to a \vee b$ iff $e_1 \wedge \neg e_2$, $\neg e_1 \wedge e_2$, $e_1 \wedge e_2$, and $\neg e_1 \wedge \neg e_2$, respectively, are propositional contradictions. Clearly, in the conjunctive (resp., disjunctive) case, the set $KB'(a, b)$ can be computed in polynomial time in the input size of e_1 and e_2, since:

- $e_1 \wedge \neg e_2$ is a propositional contradiction iff $e_1 = \bot$, $e_2 = \top$, or every basic event in e_2 appears in e_1 (resp., every basic event in e_1 appears in e_2),
- $\neg e_1 \wedge e_2$ is a propositional contradiction iff $e_2 = \bot$, $e_1 = \top$, or every basic event in e_1 appears in e_2 (resp., every basic event in e_2 appears in e_1),
- $e_1 \wedge e_2$ is a propositional contradiction iff $e_1 = \bot$ or $e_2 = \bot$,
- $\neg e_1 \wedge \neg e_2$ is a propositional contradiction iff $e_1 = \top$ or $e_2 = \top$.

Let p'_1 and p'_2 denote the probabilistic pairs $(a, [l_1, u_1])$ and $(b, [l_2, u_2])$, respectively. The claims now follow immediately from Theorem 4.15, since:

(a) $KB(e_1, e_2) \cup \{p_1, p_2\}$ is satisfiable iff $KB'(a, b) \cup \{p'_1, p'_2\}$ is satisfiable,

(b) the probabilistic conjunction (resp., disjunction, difference) of p_1 and p_2 under $KB(e_1, e_2)$ produces the same probability interval as the probabilistic conjunction (resp., disjunction, difference) of p'_1 and p'_2 under $KB'(a, b)$. $\square$

C Appendix: Operations in Relational Algebra

Suppose in the sequel that R, S etc are relation schemes on well-defined sets of attributes A_i. Then the operations of classical relational algebra can be defined as follows.

Classical *selection conditions* are inductively defined as follows. A *term* t is an attribute name A or a constant c. If t_1 and t_2 are two terms and θ belongs to $\{=, \leq\}$, then $t_1 \, \theta \, t_2$ is a selection condition. If ϕ_1 and ϕ_2 are selection conditions, then $\neg\phi_1$ and $(\phi_1 \wedge \phi_2)$ are selection conditions. The *interpretation* of a term t in a tuple w over R, denoted $[t]_w$, is defined by $[c]_w = c$ for all constants c, and $[A]_w = w(A)$ for all attribute names $A \in R$. A selection condition ϕ is *applicable* to a relation scheme R iff $[t_1]_w$, $[t_2]_w$, and $[t_1]_w \, \theta \, [t_2]_w$ are defined for every $t_1 \, \theta \, t_2$ that occurs in ϕ and every tuple w over R. Given a selection condition ϕ that is applicable to R, the *satisfaction* of ϕ in a tuple w over R, denoted $w \models \phi$, is inductively defined by $w \models t_1 \, \theta \, t_2$ iff $[t_1]_w \, \theta \, [t_2]_w$, $w \models \neg\phi_1$ iff not $w \models \phi_1$, and $w \models (\phi_1 \wedge \phi_2)$ iff $w \models \phi_1$ and $w \models \phi_2$. Given a relation r over R and a selection condition ϕ that is applicable to R, the *selection on* r *with respect to* ϕ, denoted $\sigma_\phi(r)$, is the set of all tuples v over R such that $v \in r$ and $v \models \phi$.

Given a relation r over R and a relation scheme $T \subseteq R$, the *projection of* r *to* T, denoted $\pi_T(r)$, is the set of all tuples v over T such that there are $v_r \in r$ with $v_r|_T = v$, where $v_r|_T$ is the restriction of v_r to T.

A *renaming condition* N for a relation scheme R is an expression $B_1, \ldots, B_l \leftarrow C_1, \ldots, C_l$, where $B_1, \ldots, B_l$ and $C_1, \ldots, C_l$ are sequences of distinct attribute names such that $R - \{B_1, \ldots, B_l\}$ and $\{C_1, \ldots, C_l\}$ are disjoint. The *renaming of* R (resp., a tuple v over R) *with respect to* N, denoted $\rho_N(R)$

(resp., $\rho_N(v)$), is obtained from R (resp., v) by replacing each B_i by the new attribute name C_i. Given a relation r over R and a renaming condition N for R, the *renaming of r with respect to N*, denoted $\rho_N(r)$, is the set of all tuples $\rho_N(v)$ over $\rho_N(R)$ with $v \in r$.

Given two relations r and s over R and S, respectively, the *join* of r and s, denoted $r \bowtie s$, is the set of all tuples v over $R \cup S$ such that there are tuples $v_r \in r$ and $v_s \in s$ with $v_r = v|_R$ and $v_s = v|_S$. The join of r and s for $R \cap S = \emptyset$ is also called the *Cartesian product* of r and s, denoted $r \times s$.

Given two relations r and s over R, the *union* (resp., *intersection*, *difference*) of r and s, denoted $r \cup s$ (resp., $r \cap s$, $r - s$), is the set-theoretic union (resp., intersection, difference) of r and s.

D Appendix: Proofs for Section 5

To prove the query equivalences in Section 5, we first introduce a convenient representation of pcv-relations.

We identify each pcv-relation $r = (\delta, \mu)$ of type R with the mapping ν_r from $\mathrm{dom}(R)$ to the set of all probabilistic pairs, where $\nu_r(v) = \mu(v)$, if $v \in \delta$, and $\nu_r(v) = \bot$, otherwise. Given a probability-free selection condition ϕ that is applicable to R, we associate $\sigma_\phi(r)$ with the following mapping:

$$v \quad \mapsto \quad \begin{cases} \nu_r(v) & \text{if } v \models \phi \\ \bot & \text{otherwise} \end{cases} \qquad \text{(for all } v \in \mathrm{dom}(R)\text{).}$$

A probabilistic pair $p = (e, [l, u])$ *satisfies* a selection condition ϕ of the form $x.\mathsf{lb} \geq c$ (resp., $x.\mathsf{ub} \geq c$), where x is a tuple variable and c is a rational number from $[0, 1]$, denoted $p \models \phi$, iff $l \geq c$ (resp., $u \geq c$). Given such a selection condition ϕ, we associate $\sigma_\phi(r)$ with the following mapping:

$$v \quad \mapsto \quad \begin{cases} \nu_r(v) & \text{if } \nu_r(v) \models \phi \\ \bot & \text{otherwise} \end{cases} \qquad \text{(for all } v \in \mathrm{dom}(R)\text{).}$$

Given a subtype S of R and a commutative and associative disjunction strategy $\oplus$ that has the identity property, we associate $\pi^\oplus_S(r)$ with the following mapping:

$$v \quad \mapsto \quad \bigoplus\nolimits_{w \in \mathrm{dom}(R),\, \pi_S(w) = v} \nu_r(w) \qquad \text{(for all } v \in \mathrm{dom}(S)\text{).}$$

Note that for empty (resp., infinite) $\mathrm{dom}(R)$, the expression $\bigoplus_{w \in \mathrm{dom}(R),\, \pi_S(w) = v} \nu_r(w)$ denotes

$$\bot \quad \text{(resp., } \bigoplus\nolimits_{w \in \mathrm{dom}(R),\, \pi_S(w) = v,\, \nu_r(w) \neq \bot} \nu_r(w)\text{).}$$

Given a renaming condition N for R, we associate $\rho_N(r)$ with the following mapping:

$$v \quad \mapsto \quad \nu_r(\rho_N^{-1}(v)) \qquad \text{(for all } v \in \mathrm{dom}(\rho_N(R))\text{).}$$

Given two pcv-relations r_1 and r_2 of join-compatible types R_1 and R_2, respectively, and a probabilistic conjunction strategy $\otimes$, we associate $r_1 \bowtie^\otimes r_2$ with the following mapping:

$$v \quad \mapsto \quad \nu_{r_1}(\pi_{R_1}(v)) \otimes \nu_{r_2}(\pi_{R_2}(v)) \qquad \text{(for all } v \in \mathrm{dom}(R_1 \bowtie R_2)\text{).}$$

Given two pcv-relations r_1 and r_2 of the same type R and a conjunction (resp., disjunction, difference) strategy $\oplus$ (resp., $\otimes$, $\ominus$), we associate $r_1 \cup^\oplus r_2$ (resp., $r_1 \cap^\otimes r_2$, $r_1 -^\ominus r_2$) with the following mapping:

$$v \quad \mapsto \quad \nu_{r_1}(v) \oplus \nu_{r_2}(v) \ \ \text{(resp., } \nu_{r_1}(v) \otimes \nu_{r_2}(v),\ \nu_{r_1}(v) \ominus \nu_{r_2}(v)\text{)} \quad \text{(for all } v \in \mathrm{dom}(R)\text{).}$$

In the sequel, we drop the distinction between each pcv-relation r and its associated mapping ν_r. That is, we simply write r and $r[v]$ to denote ν_r and $\nu_r(v)$, respectively.

Proof of Theorem 5.1. Equation (5.1) holds by the following relationship for each $v \in \mathrm{dom}(R)$:

$$\begin{array}{rl} & (r_1 \bowtie^{\otimes} r_2)[v] \\ = & r_1[\pi_{R_1}(v)] \otimes r_2[\pi_{R_2}(v)] \\ = & (\rho_N(r_1))[\rho_N(\pi_{R_1}(v))] \otimes r_2[\pi_{R_2}(v)] \\ = & (\rho_N(r_1))[\pi_{\rho_N(R_1)}(\rho_N(v)))] \otimes r_2[\pi_{R_2}(v)] \\ = & \bigoplus_{w \in \mathrm{dom}(\rho_N(R_1) \times R_2),\, \pi_R(w)=v,\, w \models \phi} (\rho_N(r_1))[\pi_{\rho_N(R_1)}(w)] \otimes r_2[\pi_{R_2}(w)] \\ = & \bigoplus_{w \in \mathrm{dom}(\rho_N(R_1) \times R_2),\, \pi_R(w)=v,\, w \models \phi} (\rho_N(r_1) \times^{\otimes} r_2)[w] \\ = & \bigoplus_{w \in \mathrm{dom}(\rho_N(R_1) \times R_2),\, \pi_R(w)=v} (\sigma_\phi(\rho_N(r_1) \times^{\otimes} r_2))[w] \\ = & (\pi_R^{\oplus}(\sigma_\phi(\rho_N(r_1) \times^{\otimes} r_2)))[v]\,. \quad \Box \end{array}$$

Proof of Theorem 5.2. We obtain (5.2) by the following relationship for all pcv-tuples t of type R:

$$\begin{array}{lll} t \in \sigma_{\phi_1 \wedge \phi_2}(r) & \text{iff} & t \in r \text{ and } t \models \phi_1 \wedge \phi_2 \\ & \text{iff} & (t \in r \text{ and } t \models \phi_2) \text{ and } t \models \phi_1 \\ & \text{iff} & t \in \sigma_{\phi_2}(r) \text{ and } t \models \phi_1 \\ & \text{iff} & t \in \sigma_{\phi_1}(\sigma_{\phi_2}(r))\,. \end{array}$$

Equation (5.3) follows from (5.2) and the commutativity of the conjunction in selection conditions. $\Box$

Proof of Theorem 5.3. Equation (5.4) holds by the following relationship for all $v \in \mathrm{dom}(T)$:

$$\begin{array}{rcl} (\pi_S^{\oplus}(\pi_T^{\oplus}(r)))[v] & = & \bigoplus_{w \in \mathrm{dom}(T),\, \pi_S(w)=v} (\pi_T^{\oplus}(r))[w] \\ & = & \bigoplus_{w \in \mathrm{dom}(T),\, \pi_S(w)=v} \bigoplus_{w' \in \mathrm{dom}(R),\, \pi_T(w')=w} r[w'] \\ & = & \bigoplus_{w \in \mathrm{dom}(R),\, \pi_S(w)=v} r[w] \\ & = & (\pi_S^{\oplus}(r))[v]\,. \end{array}$$

Equation (5.5) holds by the following relationship for all $v \in \mathrm{dom}(S)$:

$$\begin{array}{rcl} (\sigma_\phi(\pi_T^{\oplus}(r)))[v] & = & \begin{cases} (\pi_T^{\oplus}(r))[v] & \text{if } v \models \phi \\ \bot & \text{otherwise} \end{cases} \\ & = & \begin{cases} \bigoplus_{w \in \mathrm{dom}(R),\, \pi_T(w)=v} r[w] & \text{if } v \models \phi \\ \bot & \text{otherwise} \end{cases} \\ & = & \bigoplus_{w \in \mathrm{dom}(R),\, \pi_T(w)=v,\, v \models \phi} r[w] \\ & = & \bigoplus_{w \in \mathrm{dom}(R),\, \pi_T(w)=v,\, w \models \phi} r[w] \\ & = & \bigoplus_{w \in \mathrm{dom}(R),\, \pi_T(w)=v} (\sigma_\phi(r))[w] \\ & = & (\pi_T^{\oplus}(\sigma_\phi(r)))[v]\,. \quad \Box \end{array}$$

Proof of Theorem 5.4. We obtain (5.6) by the following relationship for all pcv-tuples t of type $\rho_N(R)$:

$$\begin{array}{lll} t \in \sigma_\phi(\rho_N(r)) & \text{iff} & t \in \rho_N(r) \text{ and } t \models \phi \\ & \text{iff} & \rho_N^{-1}(t) \in r \text{ and } \rho_N^{-1}(t) \models \phi' \\ & \text{iff} & \rho_N^{-1}(t) \in \sigma_{\phi'}(r) \\ & \text{iff} & t \in \rho_N(\sigma_{\phi'}(r))\,. \end{array}$$

Equation (5.7) holds by the following relationship for all $v \in \mathrm{dom}(S)$:

$$\begin{array}{lll} (\pi_S^{\oplus}(\rho_N(r)))[v] & = & \bigoplus_{w \in \mathrm{dom}(\rho_N(R)),\, \pi_S(w)=v} (\rho_N(r)))[w] \\ & = & \bigoplus_{w \in \mathrm{dom}(\rho_N(R)),\, \pi_S(w)=v} r[\rho_N^{-1}(w)] \\ & = & \bigoplus_{w' \in \mathrm{dom}(R),\, \pi_{S'}(w')=\rho_N^{-1}(v)} r[w'] \\ & = & (\pi_{S'}^{\oplus}(r))[\rho_N^{-1}(v)] \\ & = & (\rho_N(\pi_{S'}^{\oplus}(r)))[v]\,. \quad \Box \end{array}$$

Proof of Theorem 5.5. Equation (5.8) (resp., (5.9)) follows immediately from the commutativity (resp., associativity) of the probabilistic conjunction strategy $\otimes$.

Equation (5.10) holds by the following relationship for all $v \in \mathrm{dom}(R_1 \bowtie R_2)$:

$$\begin{array}{lll} (\sigma_{\phi_1 \wedge \phi_2}(r_1 \bowtie^{\otimes} r_2))[v] & = & \begin{cases} (r_1 \bowtie^{\otimes} r_2)[v] & \text{if } v \models \phi_1 \wedge \phi_2 \\ \bot & \text{otherwise} \end{cases} \\ & = & \begin{cases} r_1[\pi_{R_1}(v)] \otimes r_2[\pi_{R_2}(v)] & \text{if } v \models \phi_1 \wedge \phi_2 \\ \bot & \text{otherwise} \end{cases} \\ & = & (\sigma_{\phi_1}(r_1))[\pi_{R_1}(v)] \otimes (\sigma_{\phi_2}(r_2))[\pi_{R_2}(v)] \\ & = & (\sigma_{\phi_1}(r_1) \bowtie^{\otimes} \sigma_{\phi_2}(r_2))[v]\,. \end{array}$$

Equation (5.11) follows from $\pi_S^{\oplus}(r_1 \bowtie^{\otimes} r_2) = \pi_S^{\oplus}(\pi_{S_1 \bowtie S_2}^{\oplus}(r_1 \bowtie^{\otimes} r_2))$ by Equation (5.4) and the following relationship for all $v \in \mathrm{dom}(S_1 \bowtie S_2)$:

$$\begin{array}{lll} (\pi_{S_1 \bowtie S_2}^{\oplus}(r_1 \bowtie^{\otimes} r_2))[v] & = & \bigoplus_{w \in \mathrm{dom}(R_1 \bowtie R_2),\, \pi_{S_1 \bowtie S_2}(w)=v} (r_1 \bowtie^{\otimes} r_2)[w] \\ & = & \bigoplus_{w \in \mathrm{dom}(R_1 \bowtie R_2),\, \pi_{S_1 \bowtie S_2}(w)=v} r_1[\pi_{R_1}(w)] \otimes r_2[\pi_{R_2}(w)] \\ & = & (\bigoplus_{w_1 \in \mathrm{dom}(R_1),\, \pi_{S_1}(w_1)=\pi_{S_1}(v)} r_1[w_1]) \otimes \\ & & \quad (\bigoplus_{w_2 \in \mathrm{dom}(R_2),\, \pi_{S_2}(w_2)=\pi_{S_2}(v)} r_2[w_2]) \\ & = & (\pi_{S_1}^{\oplus}(r_1))[\pi_{S_1}(v)] \otimes (\pi_{S_2}^{\oplus}(r_2))[\pi_{S_2}(v)] \\ & = & (\pi_{S_1}^{\oplus}(r_1) \bowtie^{\otimes} \pi_{S_2}^{\oplus}(r_2))[v]\,. \quad \Box \end{array}$$

Proof of Theorem 5.7. Equation (5.16) (resp., (5.17)) is immediate by the commutativity (resp., associativity) of the probabilistic disjunction strategy $\oplus$. Equation (5.18) (resp., (5.19)) follows immediately from the commutativity (resp., associativity) of the probabilistic conjunction strategy $\otimes$.

Equation (5.20) holds by the following relationship for all $v \in \mathrm{dom}(R)$:

$$
\begin{aligned}
(\sigma_\phi(r_1 \cup^{\oplus} r_2))[v] \quad &= \quad \begin{cases} (r_1 \cup^{\oplus} r_2)[v] & \text{if } v \models \phi \\ \bot & \text{otherwise} \end{cases} \\
&= \quad \begin{cases} r_1[v] \oplus r_2[v] & \text{if } v \models \phi \\ \bot \oplus \bot & \text{otherwise} \end{cases} \\
&= \quad (\sigma_\phi(r_1))[v] \oplus (\sigma_\phi(r_2))[v] \\
&= \quad (\sigma_\phi(r_1) \cup^{\oplus} \sigma_\phi(r_2))[v]\,.
\end{aligned}
$$

Equations (5.21) and (5.22) can be proved analogously. Moreover, Equation (5.23) holds by the following relationship for all $v \in \mathrm{dom}(S)$:

$$
\begin{aligned}
(\pi^{\oplus}_S(r_1 \cup^{\oplus} r_2))[v] \quad &= \quad \textstyle\bigoplus_{w \in \mathrm{dom}(R),\, \pi_S(w)=v} (r_1 \cup^{\oplus} r_2)[w] \\
&= \quad \textstyle\bigoplus_{w \in \mathrm{dom}(R),\, \pi_S(w)=v} r_1[w] \oplus r_2[w] \\
&= \quad (\textstyle\bigoplus_{w \in \mathrm{dom}(R),\, \pi_S(w)=v} r_1[w]) \oplus (\textstyle\bigoplus_{w \in \mathrm{dom}(R),\, \pi_S(w)=v} r_2[w]) \\
&= \quad (\pi^{\oplus}_S(r_1))[v] \oplus (\pi^{\oplus}_S(r_2))[v] \\
&= \quad (\pi^{\oplus}_S(r_1) \cup^{\oplus} \pi^{\oplus}_S(r_2))[v]\,.
\end{aligned}
$$

Equation (5.24) holds by the following relationship for all $v \in \mathrm{dom}(R)$:

$$
(r_1 \cap^{\otimes} r_2)[v] \;=\; r_1[v] \otimes r_2[v] \;=\; r_1[\pi_R(v)] \otimes r_2[\pi_R(v)] \;=\; (r_1 \bowtie^{\otimes} r_2)[v]\,. \quad \Box
$$

Proof of Theorem 5.8. We prove only Equation (5.27). The proof of (5.25) and (5.26) is analogous.

As $\otimes$ has the bottomline property, all probabilistic pairs $p_1 = (e_1, [l_1, u_1])$ and $p_2 = (e_2, [l_2, u_2])$ satisfy the following relationship:

$$
p_1 \otimes p_2 \;\leq\; p_1 \otimes_{pc} p_2 \;=\; (e_1 \wedge e_2, [\min(l_1, l_2), \min(u_1, u_2)])\,.
$$

It thus follows $p_1 \otimes p_2 \leq p_1$ and $p_1 \otimes p_2 \leq p_2$. Hence, $p_1 \otimes p_2 \models x.\mathsf{lb} \geq c$ (resp., $p_1 \otimes p_2 \models x.\mathsf{ub} \geq c$) implies $p_1 \models x.\mathsf{lb} \geq c$ and $p_2 \models x.\mathsf{lb} \geq c$ (resp., $p_1 \models x.\mathsf{ub} \geq c$ and $p_2 \models x.\mathsf{ub} \geq c$). That is, $p_1 \otimes p_2 \models \phi$ implies $p_1 \models \phi$ and $p_2 \models \phi$. Thus, the following relationship holds for all $v \in \mathrm{dom}(R_1)$:

$$
\begin{aligned}
(\sigma_\phi(r_1 \cap^{\otimes} r_2))[v] \quad &= \quad \begin{cases} (r_1 \cap^{\otimes} r_2)[v] & \text{if } (r_1 \cap^{\otimes} r_2)[v] \models \phi \\ \bot & \text{otherwise} \end{cases} \\
&= \quad \begin{cases} r_1[v] \otimes r_2[v] & \text{if } r_1[v] \otimes r_2[v] \models \phi \\ \bot & \text{otherwise} \end{cases} \\
&= \quad \begin{cases} (\sigma_\phi(r_1))[v] \otimes (\sigma_\phi(r_2))[v] & \text{if } (\sigma_\phi(r_1))[v] \otimes (\sigma_\phi(r_2))[v] \models \phi \\ \bot & \text{otherwise} \end{cases} \\
&= \quad \begin{cases} (\sigma_\phi(r_1) \cap^{\otimes} \sigma_\phi(r_2))[v] & \text{if } (\sigma_\phi(r_1) \cap^{\otimes} \sigma_\phi(r_2))[v] \models \phi \\ \bot & \text{otherwise} \end{cases} \\
&= \quad (\sigma_\phi(\sigma_\phi(r_1) \cap^{\otimes} \sigma_\phi(r_2)))[v]\,. \quad \Box
\end{aligned}
$$

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, Reading, 1995.

[2] J. F. Baldwin. Evidential support logic programming. *Fuzzy Sets and Systems*, 24:1–26, 1987.

[3] D. Barbara, H. Garcia-Molina, and D. Porter. The management of probabilistic data. *IEEE Transactions on Knowledge and Data Engineering*, 4(5):387–502, 1992.

[4] R. Carnap. *Logical Foundations of Probability*. University of Chicago Press, Chicago, 1950.

[5] R. Cavallo and M. Pittarelli. The theory of probabilistic databases. In *Proceedings of the 13th International Conference on Very Large Databases*, pages 71–81. Morgan Kaufmann, 1987.

[6] D. Dey and S. Sarkar. A probabilistic relational model and algebra. *ACM Transactions on Database Systems*, 21(3):339–369, 1996.

[7] D. Dey and S. Sarkar. PSQL: A query language for probabilistic relational data. *Data & Knowledge Engineering*, 28:107–120, 1998.

[8] T. Eiter, J. J. Lu, T. Lukasiewicz, and V. S. Subrahmanian. Probabilistic object bases. Technical Report INFSYS RR-1843-99-11, Institut für Informationssysteme, Technische Universität Wien, 1999.

[9] T. Eiter, T. Lukasiewicz, and M. Walter. Extension of the relational algebra to probabilistic complex values. In *Proceedings of the International Symposium on Foundations of Information and Knowledge Systems (FoIKS 2000)*, volume 1762 of *LNCS*, pages 94–115. Springer, 2000.

[10] N. Fuhr and T. Rölleke. A probabilistic NF2 relational algebra for integrated information retrieval and database systems. In *Proceedings of the 2nd World Conference on Integrated Design and Process Technology*, pages 17–30. Society for Design and Process Science, 1996.

[11] N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Transactions on Information Systems*, 15(1):32–66, 1997.

[12] H. Gaifman. Concerning measures in first order calculi. *Israel Journal of Mathematics*, 2:1–18, 1964.

[13] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York, 1979.

[14] J. Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46(3):311–350, 1990.

[15] Y. Kornatzky and S. E. Shimony. A probabilistic object-oriented data model. *Data & Knowledge Engineering*, 12:143–166, 1994.

[16] H. E. Kyburg, Jr. Interval-valued probabilities. In G. de Cooman, P. Walley, and F. G. Cozman, editors, *Imprecise Probabilities Project*. 1998. Available from `http://ippserv.rug.ac.be/`.

[17] L. V. S. Lakshmanan, N. Leone, R. Ross, and V. S. Subrahmanian. ProbView: A flexible probabilistic database system. *ACM Transactions on Database Systems*, 22(3):419–469, 1997.

[18] L. V. S. Lakshmanan and F. Sadri. Probabilistic deductive databases. In *Proceedings of the International Logic Programming Symposium*, pages 254–268, 1994.

[19] T. Lukasiewicz. Local probabilistic deduction from taxonomic and probabilistic knowledge-bases over conjunctive events. *International Journal of Approximate Reasoning*, 21(1):23–61, 1999.

[20] T. Lukasiewicz. Probabilistic deduction with conditional constraints over basic events. *Journal of Artificial Intelligence Research*, 10:199–241, 1999.

[21] M. Pittarelli. An algebra for probabilistic databases. *IEEE Transactions on Knowledge and Data Engineering*, 6(2):293–303, 1994.

[22] H.-J. Schek and P. Pistor. Data structures for an integrated data base management and information retrieval system. In *Proceedings of the 8th International Conference on Very Large Data Bases*, pages 197–207. Morgan Kaufmann, 1982.

[23] D. Scott and P. Krauss. Assigning probabilities to logical formulas. In J. Hintikka and P. Suppes, editors, *Aspects of Inductive Logic*, pages 219–264. North-Holland, Amsterdam, 1966.

[24] M. Walter. An extension of relational algebra to probabilistic complex values. Master's thesis, Institut für Informatik, Universität Gießen, Germany, 1999.

[25] E. Zimányi. Query evaluation in probabilistic relational databases. *Theoretical Computer Science*, 171(1–2):179–219, 1997.