# Databases for Interval Probabilities

Wenzhong Zhao
wzhao0@cs.uky.edu
Department of Computer Science
University of Kentucky

Alex Dekhtyar
dekhtyar@cs.uky.edu
Department of Computer Science
University of Kentucky

Judy Goldsmith
goldsmit@cs.uky.edu
Department of Computer Science
University of Kentucky

**Abstract**

In today's uncertain world, imprecision in probabilistic information is often specified by probability intervals. We present here a new database framework for the efficient storage and manipulation of interval probability distribution functions and their associated contextual information. While work on interval probabilities and on probabilistic databases, has appeared before, ours is the first to combine these into a coherent and mathematically sound framework including both standard relational queries and queries based on probability theory. In particular, our query algebra allows the user not only to query existing interval probability distributions, but also to construct new ones by means of conditionalization and marginalization, as well as other more common database operations.

## 1 Introduction

Imagine that there is an election with a surprising outcome: The Rhinoceros Party has won the Senate seat and swept the local elections, contrary to all expectations, and yet the referendum on making AI conferences legal has failed, despite the fact that the Rhinoceros Party supports legalization.

What does it mean in such a case that something happens "contrary to all expectations?" Perhaps in this case, there were standard indicators that pointed to an Elephant Party win: Elephants raised more money than Rhinos and Donkeys combined; Donkeys had more yard signs; pre-election polls showed a clear lead for the Elephants, and exit polls did as well.

Perhaps an Elephant Party member wishes to file suit against the election commission, based on these irregularities. In order to do so, they must work with imprecise, probabilistic information, such as "The Money-pockets poll indicated an Elephant/Donkey/Rhino split of 62/23/10 for the senate seat, with a margin of error of 2%."

In order to show bias, the suit must exhibit a breadth of data in a wide variety of formats. Each such exhibit must be clearly labeled by its origin, format, and any underlying conditionalizing assumptions, such as "From a poll of Elephant men at their annual county pig roast and fund raiser."

Whether it is polling data or hospital records, there is always uncertainty in probabilities generated from data. And most studies used, whether for risk analysis, medical diagnosis, educational policy, or some other topic, are based on too-small data sets. There are many ways to indicate uncertainty about probabilities, the simplest of which is to replace point probabilities with intervals.

Interval probability distributions are robust, and can, in particular, take into account the possibility that probability assessments must be combined, although the relationship between them is unknown. For instance, two data sets with unknown overlap may have been used to derive these distributions, or the distributions may indicate probabilities of events not known to be (or not be) independent.

When interval probability distributions are used for reasoning, whether it is policy development or risk analysis, they should be stored in a manner easily accessible. It is best for all applications if there is a mechanism provided for performing basic probabilistic actions on the data: unions, marginalization, conditionalization, and so on. Furthermore, it is extremely useful to be able to access conditionalization information ("Only Elephants were polled for this,") or other information that helps put the interval probability distributions into the appropriate context.

We present here a database management framework that does exactly that. While the notion of operations on interval probabilities is not new (see, for instance, [20], etc.), what is new and exciting about this work is the framework for automating those operations. Our database management framework allows users to apply any of the standard operations on interval probabilities, and to reason about the resulting distributions. We automate the notion of a *path*, by which the genesis of the object is recorded. A path may specify that this distribution was obtained from data collected and this and the other time, combined using a join operation with the specified assumption on the interdependence of the two datasets.

There are several possible interpretations of interval probabilities. We choose the *possible world semantics* [8, 9, 15, 22]. This semantics captures the idea that, while exact probability distributions are not known, they are known to lie within the given intervals. Using this semantics, we introduce the extended Semistructured Probabilistic Algebra (ESP-Algebra), an analog of a relational algebra for the SPO data model. We define the operations of selection, projection, Cartesian product, join and conditionalization on SPOs and give efficient algorithms to compute them. The SPO data model and the query algebra described here provide a flexible solution for representing, storing and querying diverse probabilistic information.

In the next section, we expand on the voting-irregularities example, thus providing instances for many of the operations that are formally defined in Section 5. But first we give the basic data model in Section 3, and discuss the underlying semantics of interval probability distributions in Section 4. We then put our work into the context of other work on interval probabilities and probability databases in Section 6. Finally, we put this work into the bigger picture of Semistructured Probabilistic Databases and their algebras in Section 7.

## 2 Trouble in Sunny Hill

The town of Sunny Hill is holding elections for the mayor, State Representative and State Senator. Together with these races, residents of Sunny Hill need to vote on two ballot initiatives: whether or not to build a new park downtown and whether or not to legalize AI conferences. Candidates from three parties, Donkey, Elephant and Rhino, are vying for the elected offices and each candidate takes a position on each ballot initiative, with the Donkey party candidates generally supporting both, Elephant party candidates opposing both, and Rhino party candidate opposing the park but supporting legal AI.

The public, the candidates and their campaigns, as well as the election commissioner are kept aware of the voting trends in Sunny Hill by polls in the weeks preceding the elections. The polls are conducted among diverse groups such as representative samples of the entire town population, of likely voters, women, residents of specific neighborhoods, members of specific parties, etc. Among the questions asked on the polling surveys are current preferences of the participant for each races and about the initiatives, together with some demographic information and some supplementary questions such as whether the participant saw a specific and highly-charged infomercial.

The result of such intensive pre-election polling is that prior to the day of the election, there exists an extensive collection of polling data. Figure 1 contains examples of such data. Before discussing it, let us notice two important features of pre-election polls.

- Use of intervals. Poll results are constructed by asking a representative sample of a population a sequence of questions and then recording and, later, sorting the answers. However, each sample has a certain degree of bias, and not every participant reveals his/her true intentions. Because of this, pollsters use intervals to represent possible share of voters for each voting pattern. A typical statement is *"The straight Donkey ticket for the Senate, House and mayoral election is preferred by 30% of respondents +/- 2%"* (see Poll1 table from Figure 1). We represent such information as the interval $[28\%, 32\%]$.

- Interpretation of statistical distributions. Polling data is typically represented in tables indicating percentages of the sample that selected each specific voting pattern. Very often this statistical information is interpreted probabilistically. For example, the top line of Poll1 table in Figure 1 can be interpreted as *"The probability that a resident of Sunny Hills will vote straight Donkey ticket in the elections is between 28% and 32% based on the October 18 poll."*

Figure 1 shows a small sample of a wide variety of distributions that may be produced by the pollsters. Poll1 is a joint distribution of the expected vote for the three races based on a survey of a representative sample of the entire population of Sunny Hill taken on October 18. Poll2 contains the distribution of the vote in the mayoral race and the two ballot initiatives by men affiliated with the Donkey party who intended to vote Donkey in the Senate race, as indicated in a survey conducted on October 26. Poll3 contains information about the expected vote on the ballot initiatives by people who intended to split their vote for Senate and mayor between Donkey and Rhino parties respectively. Finally, Poll4, Poll5 and Poll6 contain information about the expected vote distribution in the mayoral race of the residents of three different parts of Sunny Hill based on the surveys taken on the same day. Sample sizes are also provided for convenience. These and other similar distributions are used by campaign managers and the elections commissioner to gain insight into the political trends in Sunny Hill. They are also collected by direct marketing associates.

Given a database of such distributions and the desire for a particular set of probabilities, how can a user access that information? Typically, polling data is stored in raw format by polling organizations, often using a relational DBMS, and is analyzed using a variety of statistical and/or mathematical packages, such as SAS, SPSS or MatLab. This software can be used to construct distributions such as those shown in Figure 1, and perform other manipulations of the data.

Neither traditional relational DBMS nor statistical software deal with storage and retrieval of the probability tables constructed during the analysis. As seen from the examples above, probability distributions are complex objects and they are hard to store in traditional relational databases. Yet we want a way to store probability tables of varying shapes and sizes, access them readily, and answer a wide variety of queries such as:

1. Find all probability distributions for voters from Downtown based on the surveys taken within two weeks of the election date;

2. Find the distribution of the mayoral vote for likely voters who plan to vote for building a new park;

3. Find all distributions in which the Donkey mayoral candidate receives more than 40% of votes.

To answer these and similar queries, the putative data repository must accept a query language capable of dealing with probability distributions and all other information associated with them as objects. In addition to that, the query language must be able to manipulate the probability distributions stored in the database and perform simple transformations of the distributions according to the laws of probability theory. For example, Query 2 (above), applied to a joint distribution of votes for mayoral race and two ballot initiatives (such as Poll2 in Figure 1), should result in the computation of a marginal probability distribution for the mayoral vote and the park ballot initiative (by excluding the second initiative from the distribution) and subsequent conditioning on park=yes.

Id: Poll1
population: entire town
date: October 18

| senate | house | mayor | $l$ | $u$ |
|---|---|---|---|---|
| Donkey | Donkey | Donkey | 28% | 32% |
| Donkey | Donkey | Elephant | 1% | 3% |
| Donkey | Donkey | Rhino | 3% | 5% |
| Donkey | Elephant | Donkey | 0% | 2% |
| Donkey | Elephant | Elephant | 4% | 6% |
| Donkey | Elephant | Rhino | 0% | 2% |
| Donkey | Rhino | Donkey | 1% | 2% |
| Donkey | Rhino | Elephant | 0% | 1% |
| Donkey | Rhino | Rhino | 2% | 5% |
| Elephant | Donkey | Donkey | 3% | 7% |
| Elephant | Donkey | Elephant | 1% | 3% |
| Elephant | Donkey | Rhino | 0% | 1% |
| Elephant | Elephant | Donkey | 3% | 5% |
| Elephant | Elephant | Elephant | 24% | 28% |
| Elephant | Elephant | Rhino | 4% | 6% |
| Elephant | Rhino | Donkey | 1% | 2% |
| Elephant | Rhino | Elephant | 0% | 3% |
| Elephant | Rhino | Rhino | 2% | 6% |
| Rhino | Donkey | Donkey | 2% | 3% |
| Rhino | Donkey | Elephant | 0% | 1% |
| Rhino | Donkey | Rhino | 1% | 3% |
| Rhino | Elephant | Donkey | 0% | 2% |
| Rhino | Elephant | Elephant | 2% | 3% |
| Rhino | Elephant | Rhino | 0% | 2% |
| Rhino | Rhino | Elephant | 2% | 4% |
| Rhino | Rhino | Elephant | 1% | 4% |
| Rhino | Rhino | Rhino | 7% | 12% |

Id: Poll2
population: Donkey men
date: October 26
senate vote: Donkey

| mayor | park | legalization | $l$ | $u$ |
|---|---|---|---|---|
| Donkey | yes | yes | 44% | 52% |
| Donkey | yes | no | 12% | 16% |
| Donkey | no | yes | 8% | 12% |
| Donkey | no | no | 4% | 8% |
| Elephant | yes | yes | 5% | 10% |
| Elephant | yes | no | 1% | 2% |
| Elephant | no | yes | 3% | 4% |
| Elephant | no | no | 6% | 8% |
| Rhino | yes | yes | 2% | 4% |
| Rhino | yes | no | 1% | 3% |
| Rhino | no | yes | 3% | 5% |
| Rhino | no | no | 1% | 4% |

Id: Poll3
population: entire town
date: October 22
senate vote: Donkey
mayor vote: Rhino

| park | legalization | $l$ | $u$ |
|---|---|---|---|
| yes | yes | 56% | 62% |
| yes | no | 14% | 20% |
| no | yes | 21% | 25% |
| no | no | 3% | 7% |

Id: Poll4
population: South Side
date: October 12
sample size: 323

| mayor | $l$ | $u$ |
|---|---|---|
| Donkey | 20% | 26% |
| Elephant | 42% | 49% |
| Rhino | 25% | 33% |

Id: Poll5
population: Downtown
date: October 12
sample size: 275

| mayor | $l$ | $u$ |
|---|---|---|
| Donkey | 48% | 55% |
| Elephant | 25% | 30% |
| Rhino | 20% | 24% |

Id: Poll6
population: West End
date: October 12
sample size: 249

| mayor | $l$ | $u$ |
|---|---|---|
| Donkey | 38% | 42% |
| Elephant | 34% | 40% |
| Rhino | 15% | 20% |

Figure 1: Polling Data for Sunny Hills elections.

In this paper, we provide a data model and query language to store, query and manipulate interval probability distribution objects. The example indicates the importance of the following features:

- probability distributions and their associated, non-probabilistic information are treated as single objects;
- probability distributions with different structure (e.g., different number/type of random variables involved) are stored in the same "relations";
- query language facilities for retrieval of full distributions based on their properties, and retrieval of parts of distributions (individual rows of the probability tables) are provided;

- query language facilities for manipulations and transformations of probability distributions according to the laws of probability theory are provided;
- *interval* probability distributions are correctly handled.

# 3 Extended Semistructured Probabilistic Object (ESPO) Data Model

In this section we extend the Semistructured Probabilistic Object (SPO) data model defined in [10] to improve the flexibility of the original semistructured data model. We will start by describing the SPO definitions from [10], after which the new, extended notion is introduced.

## 3.1 Simple Semistructured Probabilistic Objects (SPOs)

Consider a universe $\mathcal{V}$ of *random variables* $\{v'_1, \ldots, v'_m\}$. With each random variable $v \in \mathcal{V}$ we associate $dom(v)$, the set of its *possible values*. Given a set $V = \{v_1, \ldots v_q\} \subseteq \mathcal{V}$, $dom(V)$ will denote $dom(v_1) \times \ldots \times dom(v_q)$. Let $\mathcal{R} = (A_1, \ldots, A_n)$ be a collection of *regular relational attributes*. For $A \in \mathcal{R}$, $dom(A)$ will denote the domain of $A$. *Simple Semistructured Probabilistic Objects (SPOs)* are defined as follows.

**Definition 1** *A **Simple Semistructured Probabilistic Object (SPO)** $S$ is defined as a tuple $S = \langle T, V, P, C\rangle$, where*
*(i) $T = \{(A, a) | A \in \mathcal{R},\ a \in dom(A)\}$( we will refer to $T$ as the* context *of $S$);*
*(ii) $V = \{v_1, \ldots, v_q\} \subseteq \mathcal{V}$ is a set of random variables that* participate *in $S$. We require that $V \neq \emptyset$;*
*(iii) $P : dom(V) \longrightarrow [0, 1]$ is the* probability table *of $S$;*
*(iv) $C = \{(u_1, X_1), \ldots (u_s, X_s)\}$, where $\{u_1, \ldots, u_s\} = U \subseteq \mathcal{V}$ and $X_i \subseteq dom(u_i)$, $1 \leq i \leq n$, such that $V \cap U = \emptyset$. We refer to $C$ as the* set of conditionals *of $S$.*

Intuitively, a Simple Semistructured Probabilistic Object (SPO) is defined as a collection of the following four different types of information:
1. **Participating random variables.** These variables determine the probability distribution described in an SPO.
2. **Probability table.** This part of the SPO stores the actual numeric probabilities. It is convenient to visualize the probability table $P$ as a table of rows of the form $(\bar{x}, \alpha)$, where $\bar{x} \in dom(V)$ and $\alpha = P(\bar{x})$. Thus, we will speak about *rows* and *columns* of the probability table where it makes explanations more convenient.
3. **Conditionals.** A probability table may represent a *conditional distribution*, conditioned by some prior information. The conditional part of its SPO stores the prior information in one of two forms: *"random variable $u$ has value $x$"* or *"the value of random variable $u$ is restricted to a subset $X$ of its values"*. In our definition, this is represented as a pair $(u, X)$. When $X$ is a singleton set, we get the first type of condition.
4. **Context.** This part of the SPO contains supporting information for a probability distribution – information about the known values of certain parameters, which are not considered to be random variables by the application.

## 3.2 Extended Semistructured Probabilistic Objects (ESPOs)

Extended Semistructured Probabilistic Objects extend the flexibility of SPOs with a number of new features: (i) support for interval probabilities, (ii) association of context and conditionals with individual random variables and (iii) paths: information about the origins of the object. We start with formal definitions.

**Definition 2** *Let $\mathcal{R}$ be a context schema. Let $V$ be a set of random variables. An* ***extended context*** *over $\mathcal{R}$ and $V$ is a tuple $T^+ = \langle (A_1, a_1, V_1), \ldots, (A_n, a_n, V_n)\rangle$, where (i) $A_i \in \mathcal{R}$, $1 \le i \le n$; (ii) $a_i \in dom(A_i)$, $1 \le i \le n$; (iii) $V_i \subseteq \mathcal{V}$, $1 \le i \le n$.*

Intuitively, *extended context* is organized as follows. Given the context $T$ of a Simple SPO and the set of participating random variables $V$, we associate with each context value the set of random variables for which it provides additional information content.

**Example 1** *Consider the context attributes of the Simple SPO in Figure 2 (left). The* Id, population *and* date *attributes relate to the entire object. At the same time, we would like to represent the fact that 323 survey respondents indicated the intention to vote on the park construction question while 342 respondents responded to the question about their vote on the AI conference legalization. Without extended context, we can include both* responses:323 *and* responses:342 *in the SPO but we cannot associate the occurrences of the attributes with individual random variables. The SPO with extended context in the center of the Figure 2 shows how extended context alleviates this problem.*

| | | |
|---|---|---|
| Id: | Poll3 | |
| population: | entire town | |
| responses: | 323 | |
| responses: | 342 | |
| date: | October 22 | |
| park | legalization | $P$ |
| yes | yes | 0.57 |
| yes | no | 0.2 |
| no | yes | 0.25 |
| no | no | 0.08 |
| senate: | Donkey | |

| | | |
|---|---|---|
| Id: | Poll3 | |
| population: | entire town | |
| responses: | 323 {park} | |
| responses: | 342 {legalization} | |
| date: | October 22 | |
| park | legalization | $P$ |
| yes | yes | 0.57 |
| yes | no | 0.2 |
| no | yes | 0.25 |
| no | no | 0.08 |
| senate: | Donkey | |

| | | |
|---|---|---|
| Id: | Poll3 | |
| population: | entire town | |
| responses: | 323 {park} | |
| responses: | 342 {legalization} | |
| date: | October 22 | |
| park | legalization | $P$ |
| yes | yes | 0.57 |
| yes | no | 0.2 |
| no | yes | 0.25 |
| no | no | 0.08 |
| senate: | Donkey {park} | |

Figure 2: Simple vs. Extended context and conditionals in SPOs

We note that in Example 1 the other two context attributes, population and date have the scope over the entire set of random variables participating in the SPO. This can be represented explicitly, by specifying the entire set. However, we will also assume that whenever the scope is not specified for a context attribute, the scope of the attribute is the entire set of participating random variables (as we did here).

Similarly to context, we extend the conditionals.

**Definition 3** *Let $C = \{(u_1, X_1), \ldots, (u_n, X_n)\}$ be a set of* conditionals *and $V$ be a set of random variables s.t., $V \cap \{u_1, \ldots, u_n\} = \emptyset$. The set of extended conditionals $C^+$ is defined as $C^+ = \{(u_1, X_1, V_1), \ldots, (u_n, X_n, V_n)\}$, where $V_i \subseteq V$, $1 \le i \le n$.*

Extended conditionals are more subtle than extended context, but they are useful in a number of situations.

**Example 2** *To continue the previous example, consider now the conditional part of the SPO on the left side of Figure 2. The condition* senate=Donkey *applies to the entire distribution. In SPO model only such conditions can be expressed. As [10, 15] note, this leads to significant restrictions put on query algebra operations of join and Cartesian product: these two operations are defined only for pairs of SPOs with identical conditional parts. By extending conditionals to specify scope, as shown in the rightmost SPO on Figure 2, we can extend the expressive power of the framework. The particular SPO in question could have* *originated as a result of a Cartesian product (See Section 5.4) of an SPO containing information about park initiative votes of people who prefer the Donkey party candidate for the Senate and an SPO containing voter preferences on the AI conference legalization initiative. In the SPO model of [10] this operation would not have been possible.*

We can now give the definition of an Extended SPO (ESPO).

**Definition 4** *Let C[0,1] be a set of all subintervals of the interval* $[0, 1]$ *An* ***Extended Semistructured Probabilistic Object (ESPO)*** $S$ *is a tuple* $S = \langle T^+, V, P, C^+, \omega \rangle$*, where*

- $T^+$ *is extended context over some* schema $R$ *and* $V$*;*
- $V = \{v_1, \dots, v_q\} \subseteq \mathcal{V}$ *is a set of random variables that* participate *in* $S$*. We require that* $V \neq \emptyset$*;*
- $P : dom(V) \longrightarrow$ *C[0,1] is the* probability table *of* $S$*.* $P$ *must be* consistent *(see Definition 8 in Section 4);*
- $C^+ = \{(u_1, X_1, V_1), \dots, (u_n, X_n, V_n)\}$ *is a set of extended conditionals over* $V$*, and* $\{u_1, \dots, u_n\} \cap V = \emptyset$*, and*
- $\omega$*, called a* path expression *or* path *of* $S$ *is an expression in the Extended Semistructured Probabilistic Algebra.*

As mentioned above, in addition to extending context and conditionals and switching to interval probabilities, we also introduce a notion of *path* for an ESPO. Intuitively, the path on an ESPO $S$ indicates its origin. If the object was inserted into the database in its current form, then a unique id will be assigned to it. If $S$ appeared as a result of a sequence of query algebra operations, the process of constructing $S$ will be documented in its path. The exact syntax and construction of paths will be explained in Section 5.

**Example 3** *Figure 3 shows the anatomy of ESPOs. The object in the figure represents a joint probability distribution of votes in the Senate race and for the AI conference legalization ballot initiative for male voters who chose to vote Donkey for mayor. The distribution is based on the survey that took place on October 23; 238 respondents indicated their vote in the Senate race, 195 in the legalization vote, with 184 of the respondents giving both answers.*

| | | | |
|---|---|---|---|
| $\omega$: | S | | ⟵ *path expression* |
| date: | October 23 | | ⟵ *extended context* |
| gender: | male | | |
| respondents: | 238, {senate} | | |
| respondents: | 195, {legalization} | | |
| overlap: | 184 | | |
| senate | legalization | [ $l$, $u$] | ⟵ *random variables* |
| Rhino | yes | [0.04, 0.11] | |
| Rhino | no | [0.1, 0.15] | |
| Donkey | yes | [0.22, 0.27] | ⟵ *interval probability table* |
| Donkey | no | [0.09, 0.16] | |
| Elephant | yes | [0.05, 0.13] | |
| Elephant | no | [0.21, 0.26] | |
| mayor: | Donkey{senate, legalization} | | ⟵ *extended conditional* |

Figure 3: Extended Semistructured Probabilistic Object

While an ESPO consists of five components, only the first four: extended context, participating random variables, probability table and extended conditional carry the information about the distribution. The last component, the path, allows us to find the origins of a specific ESPO in the database. We note here that an ESPO with the same content of the first four components can have different paths in the database (for example because it could have originated as a results of two syntactically different but semantically equivalent queries of ESP-Algebra defined below). In many situations it is convenient not to distinguish between such ESPOs. This is formalized in the following definition.

**Definition 5** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ and $S' = \langle T^{+\prime}, V', P', C^{+\prime}, \omega'\rangle$ be two ESPOs. We say that $S$ is equivalent to $S'$, denoted $S \equiv S'$, iff $T^+ = T^{+\prime}$, $V = V'$, $P = P'$ and $C^+ = C^{+\prime}$.*

**Note:** When representing ESPOs we will assume that a lack of random variables after a context attribute or a conditional indicates that it is associated with all participating random variables. Therefore, strictly speaking, we did not need to explicitly include the list of associations for the mayor=Donkey conditional in Figure 3; we did it to make a point that the conditional part has extended syntax.

# 4 Semantics for Interval Probabilities

In [10], we assumed for simplicity that all probabilities contained in the SPOs are *point probabilities*, i.e.the probability space $\mathcal{P} = [0, 1]$. This assumption, however, is good only for the situations when we know in advance that all probabilities computed in a particular application domain will be point probabilities. There are many situations that this assumption would not hold.

- In general, when computing the probability of a conjunction of two events, knowing the point probabilities of the original events does not immediately guarantee uniqueness of the probability of the conjunction. The latter probability depends also on the known relationship between the two original events. When no such relationship is known, the probability of conjunction can only be computed to lie in an interval [4]: $\max(p(a) + p(b) - 1, 0) \leq p(a \wedge b) \leq min(p(a), p(b))$
- In some applications, it may be infeasible to obtain the exact probabilities or the point probabilities obtained will not be robust: so intervals better represent our knowledge of the domain.

In this paper we assume that the probability space is $\mathcal{P} =$ C[0,1], the set of all subintervals of the interval $[0, 1]$. The rest of this section formally introduces the possible worlds semantics for the probability distributions over $\mathcal{P}$ and the notions of *consistency* and *tightness* of the distributions. Possible worlds approach to describing interval probabilities has been adopted by a number of researchers. In particular, the semantics described here is similar to the one introduced by de Campos, Huete and Moral [8]. Similar treatment of interval probability distributions in database literature appeared in [11] where interval probability distributions were discussed in the context of Temporal Probabilistic Databases. Similar notions are also found in the work of Weichselberger [22]. In database literature, a specialized version of possible worlds semantics for interval probabilities appeared in [11]. The description of the semantics in this section follows [9]. We discuss related work in more detail in Section 6.

**Definition 6** *Let $V$ be a set of random variables. A* ***probabilistic interpretation (p-interpretation)*** *over $V$ is a function $I_V : dom(V) \to [0, 1]$, such that $\sum_{\bar{x} \in dom(V)} I_V(\bar{x}) = 1$.*

Given a set of random variables, a *p-interpretation* over it is any valid *point* probability distribution. Our main idea is that a probability distribution function (**pdf**) $P : dom(V) \to$ C[0,1] represents a set of possible point probability distributions (a.k.a., *p-interpretations*). This corresponds to de Campos, et al.'s *instance* [8].

In the rest of the paper we will adopt the following notation. Given a probability distribution $P : dom(V) \to$ C[0,1], for each $\bar{x} \in dom(V)$ we will write $P(\bar{x}) = [l_{\bar{x}}, u_{\bar{x}}]$. Whenever we enumerate $dom(V)$ as $dom(V) = \{\bar{x}_1, \ldots \bar{x}_m\}$, we will write $P(\bar{x}_i) = [l_i, u_i]$, $1 \leq i \leq m$.

**Definition 7** *Let $V$ be a set of random variables and $P : dom(V) \to$ C[0,1] a complete interval probability distribution function over $V$. A* probabilistic interpretation *$I_V$ satisfies $P$ ($I_V \models P$) iff* $(\forall \bar{x} \in dom(V))(l_{\bar{x}} \leq I_V(\bar{x}) \leq u_{\bar{x}})$.

*Let $V$ be a set of random variables and $P' : X \to$ C[0,1] an incomplete interval probability distribution function over $X \subset dom(V)$. A* probabilistic interpretation *$I_V$ satisfies $P'$ ($I_V \models P'$) iff* $(\forall \bar{x} \in X)(l_{\bar{x}} \leq I_V(\bar{x}) \leq u_{\bar{x}})$.

Basically, if a p-interpretation $I_V$ satisfies an interval probability distribution function $P$, then given $P$, $I_V$ is a *possible point probability distribution*.

**Example 4** *Consider a random variable $v$ with domain $\{a, b, c\}$. Let probability distribution functions $P_1$, $P_2$ and $P_3$ and p-interpretations $I_1$, $I_2$, $I_3$ and $I_4$ be defined in this table.*

| $P_1$ | $P_2$ | $P_3$ | $I_1$ | $I_2$ | $I_3$ | $I_4$ |
|---|---|---|---|---|---|---|
| $P_1(a) = [0.2, 0.3]$ | $P_2(a) = [0.3, 0.6]$ | $P_3(a) = [0.4, 0.5]$ | $I_1(a) = 0.3$ | $I_2(a) = 0.5$ | $I_3(a) = 0.25$ | $I_4(a) = 0.7$ |
| $P_1(b) = [0.3, 0.45]$ | $P_2(b) = [0.3, 0.4]$ | $P_3(b) = [0.4, 0.5]$ | $I_1(b) = 0.3$ | $I_2(b) = 0.4$ | $I_3(b) = 0.45$ | $I_4(b) = 0.3$ |
| $P_1(c) = [0.3, 0.5]$ | | $P_3(c) = [0.4, 0.5]$ | $I_1(c) = 0.4$ | $I_2(c) = 0.1$ | $I_3(c) = 0.3$ | $I_4(c) = 0$ |

*P-interpretation $I_1$ satisfies both $P_1$ and $P_2$. P-interpretation $I_2$ satisfies $P_2$ but not $P_1$ while $I_3$ satisfies $P_1$ but not $P_2$. Finally, $I_4$ satisfies neither $P_1$ nor $P_2$. None of the p-interpretations $I_1, I_2, I_3, I_4$ satisfies $P_3$.*

We can now specify the consistency criterion for interval probability distribution functions.

**Definition 8** *An interval probability distribution function $P : dom(V) \to$ C[0,1] is* ***consistent*** *iff there exists a p-interpretation $I_V$, such that $I_V \models P$.*

**Example 5** *Consider the interval probability distribution functions $P_1$, $P_2$ and $P_3$ described in Example 4. As that example shows, $I_1 \models P_1$ and $I_1 \models P_2$, and thus, both $P_1$ and $P_2$ are* consistent *interval probability distributions.*

*On the other hand, none of the p-interpretations from Example 4 satisfied $P_3$. One notices that any p-interpretation $I$ satisfying $P_3$ must have $I(a) \geq 0.4$, $I(b) \geq 0.4$ and $I(c) \geq 0.4$, hence $I(a) + I(b) + I(c) \geq 1.2$, which contradicts the constraint $I(a) + I(b) + I(c) = 1$ on p-interpretations. Therefore, no p-interpretation would satisfy $P_3$ and thus, $P_3$ is* inconsistent.

The following theorem specifies the necessary and sufficient conditions for an interval probability distribution function to be consistent.

**Theorem 1** *Let $V$ be a set of random variables and $P : dom(V) \to$ C[0,1] be a* complete *interval probability distribution function over $V$. Let $dom(V) = \{\bar{x}_1, \ldots, \bar{x}_m\}$ and $P(\bar{x}_i) = [l_i, u_i]$. $P$ is* ***consistent*** *iff the following two conditions hold: (1) $\sum_{i=1}^{m} l_i \leq 1$; (2) $\sum_{i=1}^{m} u_i \geq 1$.*

*Let $P' : X \to$ C[0,1] be an* incomplete *interval probability distribution function over $V$. Let $X = \{\bar{x}_1, \ldots, \bar{x}_m\}$ and $P'(\bar{x}_i) = [l_i, u_i]$. $P'$ is* ***consistent*** *iff $\sum_{i=1}^{m} l_i \leq 1$.*

Consistency is not the only property of interval probability distribution functions that is of interest. Another property of interval probability distribution functions, *tightness*, is also important.

**Example 6** *Consider the interval probability distribution $P$ as shown on Figure 4 (left). Assume that $P$ is complete. It is easy to see that $P$ is consistent (indeed, the sum of lower bound of probability intervals adds up to 0.4 and the the sum of the upper bounds adds up to 1.5). In fact, there will be many different p-interpretations satisfying $P$. Of particular interest to us are the p-interpretations that satisfy $P$ and take on marginal values. E.g., p-interpretation $I_1$: $I_1(\bar{x}_1) = 0.1; I_1(\bar{x}_2) = 0.1; I_1(\bar{x}_3) = 0.1; I_1(\bar{x}_4) = 0.7$ satisfies $P$ and hits the lower bounds of probability intervals provided by $P$ for $\bar{x}_1$, $\bar{x}_2$ and $\bar{x}_3$. Similarly, $I_2$: $I_2(\bar{x}_1) = 0.2; I_2(\bar{x}_2) = 0.2; I_2(\bar{x}_3) = 0.3; I_2(\bar{x}_4) = 0.3$; satisfies $P$ and hits the upper bounds of probability intervals for $\bar{x}_1, \bar{x}_2$ and $\bar{x}_3$. Thus, every single number in the probability intervals for $\bar{x}_1$, $\bar{x}_2$ and $\bar{x}_3$ is reachable by different p-interpretations satisfying $P$.*

*However, the same is not true for $\bar{x}_4$. If some p-interpretation $I$ satisfies $P$, then $I(\bar{x}_4) \neq 0.1$. Indeed, we know that $I(\bar{x}_1) + I(\bar{x}_2) + I(\bar{x}_3) + I(\bar{x}_4) = 1$ and if $I(\bar{x}_4) = 0.1$ then $I(\bar{x}_1) + I(\bar{x}_2) + I(\bar{x}_3) = 0.9$. However, the maximum values for $\bar{x}_1$, $\bar{x}_2$ and $\bar{x}_3$ allowed by $P$ are 0.2, 0.2 and 0.3 respectively, and they add up to only 0.7.*

*Similarly, no p-interpretation $I$ satisfying $P$ can have $I(\bar{x}_4) = 0.8$. Indeed, in this case, $I(\bar{x}_1) + I(\bar{x}_2) + I(\bar{x}_3) = 1 - 0.8 = 0.2$. However, the smallest values for $\bar{x}_1$, $\bar{x}_2$ and $\bar{x}_3$ allowed by $P$ are all 0.1 and the add up to 0.3.*

| $X$ | $l$ | $u$ |
|---|---|---|
| $\bar{x}_1$ | 0.1 | 0.2 |
| $\bar{x}_2$ | 0.1 | 0.2 |
| $\bar{x}_3$ | 0.1 | 0.3 |
| $\underline{\bar{x}_4}$ | $\underline{0.1}$ | $\underline{0.8}$ |

| $X$ | $l$ | $u$ |
|---|---|---|
| $\bar{x}_1$ | 0.1 | 0.2 |
| $\bar{x}_2$ | 0.1 | 0.2 |
| $\bar{x}_3$ | 0.1 | 0.3 |
| $\underline{\bar{x}_4}$ | $\underline{0.3}$ | $\underline{0.7}$ |

Figure 4: Tightness of interval probability distributions.

The notion of "reachability" discussed above can be formalized as follows.

**Definition 9** *Let $P : X \to$ C[0,1] be an interval probability distribution function over a set of random variables $V$. Let $X = \{\bar{x}_1, \ldots, \bar{x}_m\}$ and $P(\bar{x}_i) = [l_i, u_i]$. A number $\alpha \in [l_i, u_i]$ is **reachable** by $P$ at $\bar{x}_i$ iff there exists a p-interpretation $I_V \models P$, such that $I(\bar{x}_i) = \alpha$.*

**Proposition 1** *Let $P : X \to$ C[0,1] be an interval probability distribution function over a set of random variables $V$. If for some $\bar{x} \in X$ there exist $\alpha$, $\beta$, $l_{\bar{x}} \leq \alpha \leq \beta \leq u_{\bar{x}}$ which are both reachable by $P$ at $\bar{x}$, then any $\gamma \in [\alpha, \beta]$ is reachable by $P$ at $\bar{x}$.*

Intuitively points *unreachable* by an interval probability distribution function represent "dead weight"; they do not provide any additional information about the *possible* point probability distributions.

**Definition 10** *Let $P : X \to$ C[0,1] be an interval probability distribution over a set $V$ of random variables. $P$ is called **tight** iff* $(\forall \bar{x} \in X)(\forall \alpha \in [l_{\bar{x}}, u_{\bar{x}}])$ $\alpha$ is reachable by $P$ at $\bar{x}$.

**Example 7** *As shown in Example 6, the interval probability distribution function $P$ shown on the left-hand side of Figure 4 is* not tight. *On the other hand, interval probability distribution function $P'$ shown on the right-hand side of Figure 4 is* tight. *Its tightness follows from the fact that p-interpretations $I_1$ and $I_2$ from Example 6 both satisfy it, and now, both upper and lower bounds for $\bar{x}_4$ are reachable.*

*Function $P'$ has another important distinction w.r.t. to $P$. Indeed, one can show that for any p-interpretation $I$, $I \models P$ iff $I \models P'$, i.e., the sets of p-interpretations that satisfy $P$ and $P'$ coincide. Hence, one can say that $P'$ is a* tight equivalent *of $P$.*

We will want to replace interval probability distributions that are not tight with their *tight equivalents*. This will be done using the *tightening* operator.

**Definition 11** *Given an interval probability distribution $P$, an interval probability distribution $P'$ is its* ***tight equivalent*** *iff (i) $P'$ is tight and (ii) For each p-interpretation $I$, $I \models P$ iff $I \models P'$.*

**Proposition 2** *Each complete interval probability distribution $P$ has a unique* tight equivalent.

**Definition 12** *A* ***tightening*** *operator $\mathcal{T}$ takes as input an interval probability function $P : X \to$ C[0,1] and returns its tight equivalent $P' : X \to$ C[0,1].*

Our next goal is to compute the result of applying the tightening operator to an interval probability distribution function efficiently. First we notice that if $P$ is tight then $\mathcal{T}(P) = P$.

The theorem below specifies an efficient procedure for computing the results of tightening an interval probability distribution function.

**Theorem 2** *Let $P : dom(V) \to$ C[0,1] be a complete interval probability distribution function over a set of random variables $V$. Let $dom(V) = \{\bar{x}_1, \ldots, \bar{x}_m\}$ and $P(\bar{x}_i) = [l_i, u_i]$. Then $(\forall 1 \leq i \leq m)$*

$$(\mathcal{T}(P)(\bar{x}_i) = [\max(l_i, 1 - \sum_{j=1}^{m} u_j + u_i), \min(u_i, 1 - \sum_{j=1}^{m} l_j + l_i)].$$

In the rest of the paper we will assume that all ESPOs under consideration have *consistent* and *tight* probability distribution functions. Using the tightening operator according to Theorem 2 will allow us to replace any probability distribution function that is not tight with its tight equivalent.

**Definition 13** *An Extended Semistructured Probabilistic Object $S = \langle T^+, V, P, C^+, \omega \rangle$ is consistent iff $P$ is consistent. Also, $S$ is tight iff $P$ is tight.*

# 5 Extended Probabilistic Semistructured Algebra

In the previous two sections we have described the ESPO data model and the underlying semantics for interval probability distributions. We are now in position to define the Extended Probabilistic Semistructured Algebra (ESP-Algebra). As in [10], we will give definitions for five major operations on the objects: selection, projection, Cartesian product, join and conditionalization. In [24] we have described how these operations can be defined in a query algebra for interval probability distributions only (without context and conditionals) in a generic way. Here, we ground the operations described in [24] in the ESPO data model.

The first four operations are extensions of the standard relational algebra operations. However, these operations will be expanded significantly in comparison both with classical relational algebra [19] and with the definitions in [10]. The fifth operation, conditionalization, is specific to probabilistic databases and represents the procedure of constructing an ESPO containing a conditional probability distribution given an ESPO for some joint probability distribution. First proposed as a database operation by Dey and Sarkar [13] for a relational model with point probabilities, this operation had been extended to non-1NF databases in [10].

In the sections below, we will describe each algebra operation. We will base our examples on the elections in Sunny Hill that we have described in Section 2.

**Example 8** *Figure 5 shows different ESPOs representing a variety of polling data from Figures 1 and 3 and more. We assume that all these objects have been inserted in the database in their current form, hence, each received a unique path Id.*

| $\omega$: | $S_1$ | | | |
|---|---|---|---|---|
| gender: | men | | | |
| party: | Donkey | | | |
| date: | October 26 | | | |
| mayor | park | legalization | $l$ | $u$ |
| Donkey | yes | yes | 0.44 | 0.52 |
| Donkey | yes | no | 0.12 | 0.16 |
| Donkey | no | yes | 0.08 | 0.12 |
| Donkey | no | no | 0.04 | 0.08 |
| Elephant | yes | yes | 0.05 | 0.1 |
| Elephant | yes | no | 0.01 | 0.02 |
| Elephant | no | yes | 0.03 | 0.04 |
| Elephant | no | no | 0.06 | 0.08 |
| Rhino | yes | yes | 0.02 | 0.04 |
| Rhino | yes | no | 0.01 | 0.03 |
| Rhino | no | yes | 0.03 | 0.05 |
| Rhino | no | no | 0.01 | 0.04 |
| senate: | Donkey | | | |

| $\omega$: | $S_2$ | | |
|---|---|---|---|
| date: | October 23 | | |
| gender: | male | | |
| respondents: | 238, {senate} | | |
| respondents: | 195, {legalization} | | |
| overlap: | 184 | | |
| senate | legalization | $l$ | $u$ |
| Rhino | yes | 0.04 | 0.11 |
| Rhino | no | 0.1 | 0.15 |
| Donkey | yes | 0.22 | 0.27 |
| Donkey | no | 0.09 | 0.16 |
| Elephant | yes | 0.05 | 0.13 |
| Elephant | no | 0.21 | 0.26 |
| mayor: | Donkey{senate, legalization} | | |

| $\omega$ | $S_3$ | | |
|---|---|---|---|
| locality: | Sunny Hill | | |
| date: | October 26 | | |
| park | legalization | $l$ | $u$ |
| yes | yes | 0.56 | 0.62 |
| yes | no | 0.14 | 0.2 |
| no | yes | 0.21 | 0.25 |
| no | no | 0.03 | 0.07 |
| mayor: | Donkey | | |

| $\omega$: | $S_4$ | |
|---|---|---|
| locality: | South Side | |
| date: | October 12 | |
| sample: | 323 | |
| mayor | $l$ | $u$ |
| Donkey | 0.2 | 0.26 |
| Elephant | 0.42 | 0.49 |
| Rhino | 0.25 | 0.33 |

| $\omega$: | $S_5$ | |
|---|---|---|
| locality: | Downtown | |
| date: | October 12 | |
| sample: | 275 | |
| mayor | $l$ | $u$ |
| Donkey | 0.48 | 0.55 |
| Elephant | 0.25 | 0.3 |
| Rhino | 0.2 | 0.24 |

| $\omega$: | $S_6$ | |
|---|---|---|
| locality: | West End | |
| date: | October 12 | |
| sample: | 249 | |
| mayor | $l$ | $u$ |
| Donkey | 0.38 | 0.42 |
| Elephant | 0.34 | 0.4 |
| Rhino | 0.15 | 0.2 |

| $\omega$: | $S_7$ | |
|---|---|---|
| locality: | Sunny Hills | |
| date: | October 26 | |
| sample: | 249 | |
| mayor | $l$ | $u$ |
| Donkey | 0.33 | 0.39 |
| Elephant | 0.32 | 0.37 |
| Rhino | 0.25 | 0.3 |

Figure 5: Sunny Hill pre-election polls in ESPO format.

In a relational data model, a relation is defined as a collection of data tuples over the same set of attributes. In our model, an *Extended Semistructured Probabilistic relation* (ESP-relation) is a set of ESPOs and an *Extended Semistructured Probabilistic database* (ESP-database) is a set of ESP-relations. Grouping ESPOs into relations is done not based on structure, as is the case in the relational databases; ESPOs with different structures can co-exist in the same ESP-relation. In the examples below we will consider ESP-relation $\mathcal{S} = \{S_1, S_2, S_3, S_4, S_5, S_6, S_7\}$ consisting of ESPOs from Figure 5.

## 5.1 Selection

There is a variety of data stored in a single Extended SPO; for each individual part of the object we need to define a specific version of the selection operation, namely, selection based on context, random variables, conditionals, probabilities and probability table. The first three types of operations, described in Section 5.1.1, when applied to an ESP-relation produce a subset of that relation, but individual ESPOs do not change (except for their paths): they either satisfy the query and are returned or do not satisfy it. On the other hand, selections on probabilities or on probability tables (described in section 5.1.2) may lead to changes in the ESPOs being returned: only parts of the probability tables may "survive" such selection operations. Different types of selections are illustrated in the following example.

**Example 9** *Table 1 lists some examples of queries that should be expressible as selection queries on ESPOs. For each question we describe the desired output of the selection operation.*

Questions 1 – 3 and 5 in the example above do not involve the extensions of the SPO data model suggested in Section 3.2. To deal just with these kinds of queries, we could adapt the definitions from our original SPO algebra of [10]. Questions 4, 6 and 7, however, involve the extensions to the SPO model. Thus,

Table 1: Selection queries to ESPOs.

| # | Query | Answer |
|---|---|---|
| 1. | "What information is available about voter attitudes on October 26?" | Set of ESPOs that have date: October 26 in their context. |
| 2. | "What are other voting intentions of people who choose to vote Donkey for mayor?" | Set of ESPOs which have as a conditional mayor=Donkey. |
| 3. | "What information is known about voter intentions in the mayoral race?" | Set of ESPOs that contain mayor in the set of participating random variables |
| 4. | "What voting patterns are likely to occur with probability between 0.2 and 0.3?" | In the probability table of each ESPO, the rows with probability values guaranteed to be between 0.2 and 0.3 are found. If such rows exist, they form the probability table of the ESPO that is returned by the query. |
| 5. | "With what probability are voters likely to choose a Donkey mayor and Elephant Senator? | Set of all ESPOs that contain mayor and senate random variables, with the probability tables of each containing only the rows where mayor=Donkey and senate=Elephant. |
| 6. | "Find all distributions based on more than 200 responses about senate vote." | Set of ESPOs that contain senate random variable and responses = X with $X > 200$ is associated with it in the context. |
| 7. | "How do people who intend to vote Donkey for mayor plan to vote for the park construction ballot initiative?" | Set of ESPOs that contain park random variable and conditional mayor=Donkey is associated with it. |

the selection definitions from [10] need to be revised to incorporate new types of queries (like 6 and 7) and new formats for already defined queries (like 4).

### 5.1.1 Selection on Context, Random Variables and Conditionals

In this section, we define the selection operations that do not alter the content of the selected objects. We start by defining the acceptable languages for selection conditions for these types of selects.

Recall that the universe $\mathcal{R}$ of context attributes consists of a finite set of attributes $A_1, \ldots A_n$ with domains $dom(A_1), \ldots, dom(A_n)$. With each attribute $A \in \mathcal{R}$ we associate a set $Pr(A)$ of allowed predicates. We assume that equality and inequality are allowed for all $A \in \mathcal{R}$.

**Definition 14** *1. An **atomic context selection condition** is an expression c of the form "$A\ Q\ x\ (Q(A, x))$", where $A \in \mathcal{R}$, $x \in dom(A)$ and $Q \in Pr(A)$.*

2. *An **atomic participation selection condition** is an expression c of the form "$v \in V$", where $v \in \mathcal{V}$ is a random variable.*

3. *An **atomic conditional selection condition** is one of the following expressions: "$u = \{x_1, \ldots x_h\}$" or "$u \ni x$" where $u \in \mathcal{V}$ is a random variable and $x, x_1, \ldots, x_h \in dom(u)$. We will slightly abuse notation and write "$u = x$" instead of "$u = \{x\}$".*

4. *An **extended atomic context selection condition** is an expression $c/V$ where c is an* atomic context selection condition *and $V \subseteq \mathcal{V}$ is a set of random variables.*

5. *An **extended atomic conditional selection condition** is an expression $c/V$ where c is an* atomic conditional selection condition *and $V \subseteq \mathcal{V}$ is a set of random variables.*

**Example 10** *The table below contains some examples of selection conditions of different types for the Sunny Hill pre-election polls database.*

Table 2: Different types of conditions for selection queries.

| Selection Condition Type | Conditions |
|---|---|
| Context | $\mathsf{date} = \mathsf{October26}$; $\mathsf{sample} > 300$ |
| Extended Context | $\mathsf{respondents} > 200/\{\mathsf{senate}\}$; $\mathsf{gender} = \mathsf{men}/\{\mathsf{mayor}, \mathsf{park}\}$ |
| Participation | $\mathsf{mayor} \in V$; $\mathsf{park} \in V$ |
| Conditional | $\mathsf{mayor} = \mathsf{Donkey}$; $\mathsf{senate} \ni \mathsf{Rhino}$; $\mathsf{senate} = \{\mathsf{Rhino}, \mathsf{Donkey}\}$ |
| Extended Conditional | $\mathsf{mayor} = \mathsf{Donkey}/\{\mathsf{park}\}$; $\mathsf{senate} \ni \mathsf{Rhino}/\{\mathsf{park}, \mathsf{mayor}\}$; |

Complex selection conditions can be formed as Boolean combinations of atomic selection conditions. The definitions below formalize the selection operation on a single Extended SPO.

**Definition 15** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO and let $c = Q(A, x)$ be an atomic context selection condition. Let $S' = \langle T^+, V, P, C^+, \omega'\rangle$ where $\omega' =$ "$\sigma_c(\omega)$". Then $\sigma_c(S) = \{S'\}$ **iff** there exists a tuple $(A, a, V) \in T^+$ such that $(a^\star, x) \in Q$; otherwise $\sigma_c(S) = \emptyset$.*

**Definition 16** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO and let $c : v \in V$ be an atomic participation selection condition. Let $S' = \langle T^+, V, P, C^+, \omega'\rangle$ where $\omega' =$ "$\sigma_c(\omega)$". Then $\sigma_c(S) = \{S'\}$ **iff** $v \in V$.*

**Definition 17** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO and let $c : u = \{x_1, \ldots, x_h\}$ be an atomic conditional selection condition. Let $S' = \langle T^+, V, P, C^+, \omega'\rangle$ where $\omega' =$ "$\sigma_c(\omega)$". Then $\sigma_c(S) = \{S'\}$ **iff** $C^+ \ni (u, X)$ and $X = \{x_1, \ldots, x_h\}$.*

*Let $c : u \ni x$ be an atomic conditional selection condition. Then $\sigma_c(S) = \{S'\}$ **iff** $C^+ \ni (u, X)$ and $X \ni x$.*

**Definition 18** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO and let $c = Q(A, x)/V$ be an extended atomic context selection condition. Let $S' = \langle T^+, V, P, C^+, \omega'\rangle$ where $\omega' =$ "$\sigma_c(\omega)$". Then $\sigma_c(S) = \{S'\}$ **iff** there exists a tuple $(A, a, V) \in T^+$ such that (i) $(a^\star, x) \in Q$; (ii) $V \subseteq V^*$; otherwise $\sigma_c(S) = \emptyset$.*

**Definition 19** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO and let $c : u = \{x_1, \ldots, x_h\}/V$ be an extended atomic conditional selection condition. Let $S' = \langle T^+, V, P, C^+, \omega'\rangle$ where $\omega' =$ "$\sigma_c(\omega)$". Then $\sigma_c(S) = \{S'\}$ **iff** $C^+ \ni (u, X, V')$, $X = \{x_1, \ldots, x_h\}$, and $V \subseteq V'$.*

*Let $c : u \ni x/$ be an extended atomic conditional selection condition. Then $\sigma_c(S) = \{S'\}$ **iff** $C^+ \ni (u, X, V')$, $X \ni x$ and $V \subseteq V'$.*

The semantics of atomic selection conditions discussed so far can be extended to their Boolean combinations in a straightforward manner: $\sigma_{C \wedge C'}(S) = \sigma_C(\sigma_{C'}(S))$ and $\sigma_{C \vee C'}(S) = \sigma_C(S) \vee \sigma_{C'}(S)$.

The interpretation of *negation* in the context selection condition requires some additional explanation. In order for a selection condition of the form $\neg Q(A, x)$ to succeed on an ESPO $S = \langle T^+, V, P, C^+, \omega\rangle$, attribute $A$ must be present in $T^+$. If $A$ is not present in the context of $S$, the selection condition does not get evaluated and the result will be $\emptyset$. Therefore, the statement $S \in \sigma_c(S) \vee S \in \sigma_{\neg c}(S)$ is not necessarily true. This also applies to conditional selection conditions.

Finally, for an ESP-relation $\mathcal{S}$, $\sigma_C(\mathcal{S}) = \bigcup_{S \in \mathcal{S}}(\sigma_C(S))$.

We note here that, whenever an ESPO satisfies any of the selection conditions described above, four of its five components, namely, context, participating variables, probability table and conditional are returned intact. The only part of the ESPO that changes is its path: the new path expression reflects the fact that the selection query had been applied to the object.

**Example 11** *Consider our ESP-relation $\mathcal{S}$ (Figure 5). Below are some possible queries to this relation and their results (we specify the unique ids of the ESPOs that match the query).*

| Id | Type | Query | Result |
|---|---|---|---|
| Q1 | context | $\sigma_{\text{date}=\text{October26}}(\mathcal{S})$ | $\{S_1, S_3, S_7\}$ |
| Q2 | participation | $\sigma_{\text{mayor}\in V}(\mathcal{S})$ | $\{S_1, S_4, S_5, S_6, S_7\}$ |
| Q3 | conditionals | $\sigma_{\text{senate}=\{\text{Donkey}\}}(\mathcal{S})$ | $\{S_1, S_3\}$ |
| Q4 | ext. context | $\sigma_{\text{respndents}>200/\{\text{senate}\}}(\mathcal{S})$ | $\{S_2\}$ |
| Q5 | ext. context | $\sigma_{\text{gender}=\text{men}/\{\text{mayor,park}\}}(\mathcal{S})$ | $\{S_1\}$ |
| Q6 | ext. conditional | $\sigma_{\text{mayor}=\{\text{Donkey}\}/\{\text{senate}\}}(\mathcal{S})$ | $\{S_2\}$ |
| Q7 | ext. conditional | $\sigma_{\text{mayor}=\{\text{Donkey}\}/\{\text{senate,house}\}}(\mathcal{S})$ | $\emptyset$ |

### 5.1.2 Selection on Probabilities and Probability Tables

The two types of selections introduced in this section are more complex. The result of a selection operation of either type depends on the content of the probability table, which can be considered as a relation (each row being a single record). In the process of performing the probabilistic selection or selection on the probability table (see questions 4 and 5, Example 9, respectively), each row of the probability table is examined individually to determine whether it satisfies the selection condition. It is retained in the answer if it does and is thrown out if it does not. Thus, a possible result of either of these two types of selection operation is an ESPO with an *incomplete* probability table. As the selection condition relates only to the content of the probability table of an ESPO, its context, participating random variables, and conditionals are preserved. We start by defining selection on probability tables.

**Definition 20** *An* ***atomic probabilistic table selection condition*** *is an expression of the form $v = x$ where $v \in \mathcal{V}$ and $x \in dom(v)$.* ***Probabilistic table selection conditions*** *are Boolean combinations of atomic probabilistic table selection conditions.*

**Definition 21** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO, $V = \{v_1, \ldots, v_k\}$, and let $c : v = x$ be an atomic probabilistic table selection condition.*

*If $v \in V$, then (assuming $v = v_i, 1 \leq i \leq k$) the result of selection from $S$ on $c$, $\sigma_c(S)$ is a semistructured probabilistic object $S' = \langle T^+, V, P', C^+, \omega'\rangle$, where $\omega' =$ "$\sigma_c(\omega)$" and*

$$P'(y_1, \ldots, \mathbf{y_i}, \ldots, y_k) = \begin{cases} P(y_1, \ldots, \mathbf{y_i}, \ldots, y_k) & \text{if } \mathbf{y_i} = x; \\ \textbf{\textit{undefined}} & \text{if } \mathbf{y_i} \neq x. \end{cases}$$

**Example 12** *Consider the ESPO $S_1$ from Figure 5. The leftmost ESPO of Figure 6 shows the result of the selection query on probability table: $\sigma_{\text{park}=\text{yes}}(S_1)$ (find the probability of all voting outcomes where respondents support the park ballot initiative). Following Definition 21, the result of this query is computed as follows: the context, list of conditionals and participating random variables remain the same, while the probability table now contains only the rows that satisfy the selection condition and the path changes to reflect the selection operation.*

*We note that if the same query is applied to the entire relation $\mathcal{S}$, the resulting relation will contain two ESPOs constructed from $S_1$ and $S_3$: only those ESPOs have participating random variable* park *(and rows for* park=yes*).*

We are now ready to describe the last type of the selection operation: selection on probabilities.

**Example 13** *The following queries are examples of the types of probabilistic selection queries that need to be expressible in ESP-Algebra.*

1. *Find all rows where the* lower bound *is equal to 0.1;*

2. *Find all rows where the* upper bound *is greater than 0.4;*
3. *Find all rows where the* probability *is* ***guaranteed*** *to be greater than 0.2;*
4. *Find all rows where the* probability ***can be*** *less than 0.2.*

| $\omega$: | $\sigma_{\mathsf{path=yes}}(S_1)$ | | | |
|---|---|---|---|---|
| gender: | men | | | |
| party: | Donkey | | | |
| date: | October 26 | | | |
| mayor | park | legalization | $l$ | $u$ |
| Donkey | yes | yes | 0.44 | 0.52 |
| Donkey | yes | no | 0.12 | 0.16 |
| Elephant | yes | yes | 0.05 | 0.1 |
| Elephant | yes | no | 0.01 | 0.02 |
| Rhino | yes | yes | 0.02 | 0.04 |
| Rhino | yes | no | 0.01 | 0.03 |
| senate: | Donkey | | | |

| $\omega$: | $\sigma_{u>0.14}(S_2)$ | | |
|---|---|---|---|
| date: | October 23 | | |
| gender: | male | | |
| respondents: | 238, {senate} | | |
| respondents: | 195, {legalization} | | |
| overlap: | 184 | | |
| senate | legalization | $l$ | $u$ |
| Rhino | no | 0.1 | 0.15 |
| Donkey | yes | 0.22 | 0.27 |
| Donkey | no | 0.09 | 0.16 |
| Elephant | no | 0.21 | 0.26 |
| mayor: | Donkey{senate, legalization} | | |

| $\omega$: | $\sigma_{l<0.11}(S_2)$ | | |
|---|---|---|---|
| date: | October 23 | | |
| gender: | male | | |
| respondents: | 238, {senate} | | |
| respondents: | 195, {legalization} | | |
| overlap: | 184 | | |
| senate | legalization | $l$ | $u$ |
| Rhino | yes | 0.04 | 0.11 |
| Rhino | no | 0.1 | 0.15 |
| Donkey | no | 0.09 | 0.16 |
| Elephant | yes | 0.05 | 0.13 |
| mayor: | Donkey{senate, legalization} | | |

Figure 6: Selection on probability table and probabilities.

The first two queries refer to the lower and upper bounds as supplied by the $P$ function. The last two queries refer to the point probability value as associated with a row by a p-interpretation. The third query specifies a for-all condition, which is true iff the condition is true for **all** p-interpretations satisfying $P$. The fourth query specifies an exists condition, which is true if at least one p-interpretation satisfying $P$ is satisfies the condition. Our constraint language will allow for all four types of atomic conditions to be expressed.

**Definition 22** *An* ***atomic probabilistic selection condition*** *is an expression of one of the forms: (i) $l$ op $\alpha$; (ii) $u$ op $\alpha$; (iii) $\forall P$ op $\alpha$; (iv) $\exists P$ op $\alpha$, where $\alpha \in [0,1]$ and op $\in \{=, \neq, \leq, \geq, <, >\}$.* ***Probabilistic selection conditions*** *are Boolean combinations of* atomic probabilistic selection conditions.

**Example 14** *While the precise semantics of probabilistic selection conditions is determined in Definition 23 below, the following conditions match the probabilistic queries from Example 13: (1) $l = 0.1$; (2) $u > 0.4$; (3) $\forall P > 0.2$; (4) $\exists P < 0.2$.*

**Definition 23** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO. Let $c$ : $l$ op $\alpha$ ($c$ : $u$ op $\alpha$) be a probabilistic atomic selection condition. Let $\bar{x} \in dom(V)$. The result of selection from $S$ on $c$ is defined as follows: $\sigma_{P\ \mathsf{op}\ \alpha}(S) = S' = \langle T^+, V, P', C^+, \omega'\rangle$, where $\omega' =$ "$\sigma_c(\omega)$" and*

$$P'(\bar{x}) = \begin{cases} P(\bar{x}) & \text{if } l_{\bar{x}}\ \mathsf{op}\ \alpha\ (u_{\bar{x}}\ \mathsf{op}\ \alpha); \\ \textbf{undefined} & \text{otherwise.} \end{cases}$$

*Let $c$ : $\forall P$ op $\alpha$ be a probabilistic atomic selection condition. The result of selection from $S$ on $c$ is defined as follows: $\sigma_{P\ \mathsf{op}\ \alpha}(S) = S' = \langle T^+, V, P', C^+, \omega'\rangle$, where $\omega' =$ "$\sigma_c(\omega)$" and*

$$P'(\bar{x}) = \begin{cases} P(\bar{x}) & \text{if } (\forall\, I \models P)(I(\bar{x})\ \mathsf{op}\ \alpha); \\ \textbf{undefined} & \text{otherwise.} \end{cases}$$

*Let $c$ : $\exists P$ op $\alpha$ be a probabilistic atomic selection condition. The result of selection from $S$ on $c$ is defined as follows: $\sigma_{P\ \mathsf{op}\ \alpha}(S) = S' = \langle T^+, V, P', C^+, \omega'\rangle$, where $\omega' =$ "$\sigma_c(\omega)$" and*

$$P'(\bar{x}) = \begin{cases} P(\bar{x}) & \text{if } (\exists\, I \models P)(I(\bar{x})\ \mathsf{op}\ \alpha); \\ \textbf{undefined} & \text{otherwise.} \end{cases}$$

**Example 15** *The center and the rightmost ESPOs on Figure 6 represent the results of selections on probabilities: $\sigma_{u>0.14}(S_2)$ and $\sigma_{l<0.11}(S_2)$ respectively. In both cases, the results of the selection keep the same context, conditionals and participating random variables, while the probability table is modified to retain only the rows where the upper (lower) bound on the probability interval satisfies the selection condition.*

*Notice that the result of $\sigma_{u>0.14}(\mathcal{S})$ would contain seven ESPOs: every object in $\mathcal{S}$ contains rows where upper bound on probability is greater that 0.14. The result of $\sigma_{l<0.11}(\mathcal{S})$ will contain two ESPOs constructed from $S_1$ and $S_2$: only those had rows with lower probability less than 0.11.*

While evaluation of the probabilistic selection conditions on lower and upper bounds is fairly straightforward, evaluation of the probabilistic selection conditions referring to p-interpretations may seem to be complex. As it turns out, these conditions can be expressed via the conditions on upper and lower bounds as specified in the following proposition.

**Proposition 3** *The following equivalences hold:*

| $\forall$-conditions | $\exists$-conditions |
|---|---|
| $\sigma_{(\forall P=\alpha)}(\mathcal{S}) \equiv \sigma_{l=\alpha\wedge u=\alpha}(\mathcal{S})$ | $\sigma_{(\exists P=\alpha)}(\mathcal{S}) \equiv \sigma_{l\leq\alpha\wedge u\geq\alpha}(\mathcal{S})$ |
| $\sigma_{(\forall P\geq\alpha)}(\mathcal{S}) \equiv \sigma_{l\geq\alpha}(\mathcal{S})$ | $\sigma_{(\exists P\geq\alpha)}(\mathcal{S}) \equiv \sigma_{u\geq\alpha}(\mathcal{S})$ |
| $\sigma_{(\forall P>\alpha)}(\mathcal{S}) \equiv \sigma_{l>\alpha}(\mathcal{S})$ | $\sigma_{(\exists P>\alpha)}(\mathcal{S}) \equiv \sigma_{u>\alpha}(\mathcal{S})$ |
| $\sigma_{(\forall P\leq\alpha)}(\mathcal{S}) \equiv \sigma_{u\leq\alpha}(\mathcal{S})$ | $\sigma_{(\exists P\leq\alpha)}(\mathcal{S}) \equiv \sigma_{l\leq\alpha}(\mathcal{S})$ |
| $\sigma_{(\forall P<\alpha)}(\mathcal{S}) \equiv \sigma_{u<\alpha}(\mathcal{S})$ | $\sigma_{(\exists P<\alpha)}(\mathcal{S}) \equiv \sigma_{l<\alpha}(\mathcal{S})$ |
| $\sigma_{(\forall P\neq\alpha)}(\mathcal{S}) \equiv \sigma_{l>\alpha\vee u<\alpha}(\mathcal{S})$ | $\sigma_{(\exists P\neq\alpha)}(\mathcal{S}) \equiv \sigma_{l\neq\alpha\vee u\neq\alpha}(\mathcal{S})$ |

**Example 16** *Figure 7 shows some selections on probabilities that use p-interpretation notation. First query, $\sigma_{\forall P<0.11}(S_1)$ finds all rows in the probability table of $S_1$ for which all p-interpretations have probability less than 0.11. By Proposition 3, this query is equivalent to $\sigma_{u<0.11}(S_1)$. Second query, $\sigma_{\exists P<0.04}(S_1)$ asks for rows of the probability table of $S_1$ in which at least one satisfying p-interpretation can have probability less than or equal to 0.04. By Proposition 3, it is equivalent to $\sigma_{l\leq 0.04}(S_1)$.*

| $\omega$: | $\sigma_{\forall P<0.11}(S_1)$ | | | |
|---|---|---|---|---|
| gender: | men | | | |
| party: | Donkey | | | |
| date: | October 26 | | | |
| mayor | park | legalization | $l$ | $u$ |
| Donkey | no | no | 0.04 | 0.08 |
| Elephant | yes | yes | 0.05 | 0.1 |
| Elephant | yes | no | 0.01 | 0.02 |
| Elephant | no | yes | 0.03 | 0.04 |
| Elephant | no | no | 0.06 | 0.08 |
| Rhino | yes | yes | 0.02 | 0.04 |
| Rhino | yes | no | 0.01 | 0.03 |
| Rhino | no | yes | 0.03 | 0.05 |
| Rhino | no | no | 0.01 | 0.04 |
| senate: | Donkey | | | |

| $\omega$: | $\sigma_{\exists P<0.04}(S_1)$ | | | |
|---|---|---|---|---|
| gender: | men | | | |
| party: | Donkey | | | |
| date: | October 26 | | | |
| mayor | park | legalization | $l$ | $u$ |
| Donkey | no | no | 0.04 | 0.08 |
| Elephant | yes | no | 0.01 | 0.02 |
| Elephant | no | yes | 0.03 | 0.04 |
| Rhino | yes | yes | 0.02 | 0.04 |
| Rhino | yes | no | 0.01 | 0.03 |
| Rhino | no | yes | 0.03 | 0.05 |
| Rhino | no | no | 0.01 | 0.04 |
| senate: | Donkey | | | |

Figure 7: Probabilistic selection on $\exists P$ and $\forall P$ conditions.

Different selection operations (both described in this section and in Section 5.1.1) commute, as shown in the following theorem:

**Theorem 3** *Let c and c′ be two selection conditions and let $\mathcal{S}$ be a semistructured probabilistic relation. Then $\sigma_c(\sigma_{c'}(\mathcal{S}) = \sigma_{c'}(\sigma_c(\mathcal{S}))$.*

## 5.2 Projection

Projection in classical relational algebra removes columns from the relation and, if needed, collapses duplicate tuples. ESPOs consist of four different components that can be affected by projection operation. We distinguish between three different types of projection here: on context, on conditionals and on participating random variables, the latter, affecting probability table as well.

There are two issues that need to be addressed when defining projection on context. First, contexts may contain numerous copies of relational attributes. Hence, projecting out a particular attribute from a context of an ESPO should result in *all* copies if this attribute being projected out. The second issue is the fact that in extended context, different attributes are associated with different participating random variables. Thus, it would be desirable to be able to take these associations into account when performing projections.

To address these two issues we define two types of projection on context. The first operation will be similar to standard relational projection, while the second operation will work by removing associations between context attributes and random variables.

**Definition 24** *Let* $F = \{A_1, \dots, A_k\}$ *be a set of* context attributes *and* $S = \langle T^+, V, P, C^+, \omega\rangle$ *be an ESPO.* ***Projection of*** $S$ ***on*** $F$*, denoted* $\pi_F(S)$ *is an ESPO* $S' = \langle T^{+'}, V, P, C^+, \omega'\rangle$*, where* $T^{+'} = \{(A, a, V^*) | (A, a, V^*) \in T^+,\ A \in F\}$ *and* $\omega' =$ "$\pi_F(\omega)$".

**Definition 25** *Let* $F^+ = \{(A_1, V_1) \dots, (A_k, V_k)\}$ *be a set of pairs where for* $1 \leq i \leq k$*,* $A_i$ *a context attribute and* $V_i \subseteq \mathcal{V}$*. Let* $S = \langle T^+, V, P, C^+, \omega\rangle$ *be an ESPO.* ***Projection of*** $S$ ***on*** $F^+$*, denoted* $\pi_{F^+}(S)$ *is an ESPO* $S' = \langle T^{+'}, V, P, C^+, \omega'\rangle$*, where* $T^{+'} = \{(A, a, V') | (A, a, V^*) \in T^+,\ A = A_i \in \{A_1, \dots A_k\},$ for some $1 \leq i \leq k,$ and $\emptyset \neq V' = V^* \cap V_i\}$ *and* $\omega' =$ "$\pi_{F^+}(\omega)$".

Given an ESPO $S$ and a set of pairs $F^+$ as described in Definition 25, the projection operation will proceed as follows. The set of context attributes *to keep* which comes from $F^+$ specifies for each attribute the list of random variables for which it is allowed to be kept. The projection operation (i) removes from the input ESPO $S$ all attributes not in $F^+$ and (ii) for each instance $(a, V^*) \in T^+$ of attribute $A_i$ s.t., $(A_i, V_i) \in F^+$ it will remove all references in $V^*$ that are not in $V_i$. If $V^* \cap V_i = \emptyset$, then $(a, V^*)$ is omitted from the projection. Projections on context are illustrated in the example below.

**Example 17** *Figure 8 contains the results of two projection queries applied to the ESPO* $S_3$ *from Figure 5,* $\pi_{\mathsf{date}}(S_3)$ *(left) and* $\pi_{\{\mathsf{locality}/\{\mathsf{park}\},\mathsf{date}/\{\mathsf{park}\}\}}(S_3)$ *(right). The first operation results in the removal of all context attributes other than* date *from the context of the ESPO. The second operation removes associations between the context attributes* locality *and* date *and all random variables but* park*. In both cases, conditionals, participating random variables and probability table are not affected.*

| $\omega$ | $\pi_{\mathsf{date}}(S_3)$ | | |
|---|---|---|---|
| date: | October 26 | | |
| park | legalization | $l$ | $u$ |
| yes | yes | 0.56 | 0.62 |
| yes | no | 0.14 | 0.2 |
| no | yes | 0.21 | 0.25 |
| no | no | 0.03 | 0.07 |
| senate: | Donkey | | |

| $\omega$ | $\pi_{\{\mathsf{locality}/\{\mathsf{park}\},\mathsf{date}/\{\mathsf{park}\}\}}(S_3)$ | | |
|---|---|---|---|
| locality: | Sunny Hill {park} | | |
| date: | October 26 {park} | | |
| park | legalization | $l$ | $u$ |
| yes | yes | 0.56 | 0.62 |
| yes | no | 0.14 | 0.2 |
| no | yes | 0.21 | 0.25 |
| no | no | 0.03 | 0.07 |
| senate: | Donkey | | |

Figure 8: Projections on context.

Projection operations on conditionals can be defined similarly. We note here that, while syntactically these operations are similar, projecting conditionals out of ESPOs is a more dangerous operation from the probability theory point of view. Basically, it is taking a conditional probability distribution $P(Y|X)$ and making a decision to “forget” the conditioning information $X$, and refer to the distribution as $P(Y)$ from then on. If unconditional distribution $P(Y)$ is also available, this may lead to some confusion. However, in some cases, projecting out conditionals is meaningful: when a specific random variable is removed from consideration in the entire database, it needs to be projected out from both lists of participating random variables (see below) as well as from the conditionals of affected ESPOs. Similarly, if the users switch to a sub-database, all ESPOs in which contain the same conditional part, that conditional part can be removed for convenience. With this in mind we present the projection on conditionals.

**Definition 26** *Let $U = \{u_1, \ldots, u_k\} \subseteq \mathcal{V}$ be a set of* random variables *and $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO.* ***Projection of** $S$ **on** $U$, denoted $\pi_{C:U}(S)$*[1] *is an ESPO $S' = \langle T^+, V, P, C^{+'}, \omega'\rangle$, where $C^{+'} = \{(u, X, V^*) | (u, X, V^*) \in C^+, \text{ and } u \in U\}$ and $\omega' =$ “$\pi_{C:U}(\omega)$”.*

**Definition 27** *Let $U^+ = \{(u_1, V_1) \ldots, (u_k, V_k)\}$ be a set of pairs where for all $1 \leq i \leq k$, $u_i \in \mathcal{V}$ and $V_i \subseteq \mathcal{V}$. Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO.* ***Projection of** $S$ **on** $U^+$, denoted $\pi_{C:U^+}(S)$ is an ESPO $S' = \langle T^+, V, P, C^{+'}, \omega'\rangle$, where $C^{+'} = \{(u, X, V') | (u, X, V^*) \in T^+,\ u = u_i \in \{u_1, \ldots, u_k\}$, and $\emptyset \neq V' = V^* \cap V_i\}$; $\omega' =$ “$\pi_{C:U}(\omega)$”.*

The following example illustrates how projection operations on conditionals work.

**Example 18** *Figure 9 shows the results of two different projection on conditionals operations, $\pi_{\mathsf{C}:\emptyset}(S_2)$ (left) and $\pi_{\mathsf{C}:\{\mathsf{mayor}/\{\mathsf{senate}\}\}}(S_2)$ (right). The first projection removes* all *conditionals in $S_2$, leaving the conditional component empty. The second projection severs the association between the* mayor = Donkey *conditional and the random variable* legalization*. Context, random variables and probability table are left intact.*

| $\omega$: | $\pi_{\mathsf{C}:\emptyset}(S_2)$ | | |
|---|---|---|---|
| date: | October 23 | | |
| gender: | male | | |
| respondents: | 238, {senate} | | |
| respondents: | 195, {legalization} | | |
| overlap: | 184 | | |
| senate | legalization | $l$ | $u$ |
| Rhino | yes | 0.04 | 0.11 |
| Rhino | no | 0.1 | 0.15 |
| Donkey | yes | 0.22 | 0.27 |
| Donkey | no | 0.09 | 0.16 |
| Elephant | yes | 0.05 | 0.13 |
| Elephant | no | 0.21 | 0.26 |

| $\omega$: | $\pi_{\mathsf{C}:\{\mathsf{mayor}/\{\mathsf{senate}\}\}}(S_2)$ | | |
|---|---|---|---|
| date: | October 23 | | |
| gender: | male | | |
| respondents: | 238, {senate} | | |
| respondents: | 195, {legalization} | | |
| overlap: | 184 | | |
| senate | legalization | $l$ | $u$ |
| Rhino | yes | 0.04 | 0.11 |
| Rhino | no | 0.1 | 0.15 |
| Donkey | yes | 0.22 | 0.27 |
| Donkey | no | 0.09 | 0.16 |
| Elephant | yes | 0.05 | 0.13 |
| Elephant | no | 0.21 | 0.26 |
| mayor: | Donkey{senate} | | |

Figure 9: Projections on conditionals.

Now we are ready to define the most intricate projection operation, projection on the set of random variables. When defining this operation, we need to keep in mind the following: (i) projection is only allowed

[1] Symbol “C” is used in the notation to distinguish the projection operation from the projection on the set of participating random variables, to be defined below.

if at least one random variable remains in the resulting set of participating random variables [2]; (ii) projecting out a random variable $v$ should result in removal of $v$ from the extended context and conditionals; (iii) projecting out a random variable $v$ should remove this variable from the probability table, i.e. the underlying probability distribution function will change. Out of these notes, the last is of the most importance.

**Definition 28** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO, and let $V^* \subset \mathcal{V}$.* ***Projection of** $S$ **on** $V^*$, denoted $\pi_{V^*}(S)$ is defined as follows:*

1. $V^* \cap V = \emptyset$: $\pi_{V^*}(S) = \emptyset$.
2. $V^* \cap V = V' \neq \emptyset$: $\pi_{V^*}(S) = S' = \langle T^{+'}, V', P', C^{+'}, \omega'\rangle$, *where*
   - $T^{+'} = \{(A, a, W') | (A, a, W) \in T^+ \text{ and } W \cap V^* \neq \emptyset \text{ and } W' = W \cap V^*\}$
   - $C^{+'} = \{(u, X, W') | (u, X, W) \in C^+ \text{ and } W \cap V^* \neq \emptyset \text{ and } W' = W \cap V^*\}$
   - $P' : dom(V') \to$ ***C[0,1]****.*
     *For all $\bar{x'} \in dom(V')$ and $(\bar{x'}, \bar{x''}) \in dom(V)$,*
     $$P'(\bar{x'}) = [\min_{I \models P}(\sum_{(\bar{x'},\bar{x''}) \in dom(V)} I(\bar{x'}, \bar{x''}))\ , \max_{I \models P}(\sum_{(\bar{x'},\bar{x''}) \in dom(V)} I(\bar{x'}, \bar{x''}))].$$
   - $\omega' =$ "$\pi_{V^*}(\omega)$"

This definition requires a careful explanation. Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO, and let $V^* \subseteq \mathcal{V}$ be the set of projection random variables. The computation of $\pi_{V^*}(S)$ proceeds as follows. First, we check if the intersection of $V$, the set of participating random variables of $S$ and $V^*$ is empty, and if it is, we return empty set as the answer. If $V' = V \cap V^*$ is not empty, we build the projection as follows:

(i) the new set of participating random variables is $V'$;

(ii) the new context $T^{+'}$ and conditionals $C^{+'}$ are produced from $T^+$ and $C^+$ respectively, by eliminating all random variables *not from* $V'$ from the extensions (associations). Context entries (conditionals) from $T^+$ ($C^+$) associated only with variables *not from* $V'$ will be eliminated from $T^{+'}$ ($C^{+'}$);

(iii) finally, the new probability table function is defined as follows. The function must range over $dom(V')$. As $V' \subseteq V$, with each value $\bar{x'} \in dom(V')$, a set of values $(\bar{x'}, \bar{x''}) \in dom(V)$ is associated, where $\bar{x''}$ ranges over $dom(V - V')$. Given a p-interpretation $I \models P$, for each $\bar{x'} \in dom(V')$ we can compute the probability assigned to it by $P$ as $I(\bar{x'}) = \sum_{\bar{x''} \in dom(V-V')} I(\bar{x'}, \bar{x''})$. Now, we know that the probability of $\bar{x'}$ has to range between the minimal and maximal value of $I(\bar{x'})$, for all $I \models P$. This interval, $[\min_{I \models P} I(\bar{x'}), \max_{I \models P} I(\bar{x'})]$ is defined to be the value of the new probability distribution function $P'$ on $\bar{x'}$.

While the computation of the new set of participating random variables, context and conditionals according to Definition 28 is straightforward, computing the new probability table requires solving a number of optimization problems (finding mins and maxs of $\sum I(\bar{x'}, \bar{x''})$ for all $\bar{x'}$), which seems like a fairly tedious task. However, it turns out that these optimization problems have analytical solutions.

[2]We want our query algebra to be closed: ESPOs in — ESPOs out. Removing all random variables from the ESPO basically collapses it. The object returned by such an operation will no longer satisfy our definition of an ESPO. Because of that, we do not consider such operations.

**Theorem 4** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO and $V^* \subseteq \mathcal{V}$. Let $V \cap V^* \neq \emptyset$ and $S' = \langle T^{+'}, V', P', C^{+'}, \omega'\rangle = \pi_{V^*}(S)$. Let $P''(x') = [\sum_{(\bar{x'},\bar{x''}) \in dom(V)} l_{(\bar{x'},\bar{x''})}, \min(1, \sum_{(\bar{x'},\bar{x''}) \in dom(V)} u_{(\bar{x'},\bar{x''})})]$. Then, $P' = \mathcal{T}^+(P'')$.*

The projection on the set of random variables is illustrated in the example below.

**Example 19** *Figure 10 illustrates the process of computing projection $\pi_{\{\mathsf{senate}\}}(S_2)$ on participating random variables. The first step of this operation is the removal of all other random variables from the probability table. Next, the duplicate rows of the new probability table are collapsed and the probability intervals are added. After that, the operation on tightening is performed to find the true intervals as we can observe that neither of the three lower bounds is reachable. We then exclude* respondents:195 *from the context as it is not associated with* senate *variable and disassociate* legalization *with conditionals.*

| $\omega$: | $S_2$ | | |
|---|---|---|---|
| date: | October 23 | | |
| gender: | male | | |
| respondents: | 238, {senate} | | |
| respondents: | 195, {legalization} | | |
| overlap: | 184 | | |
| senate | legalization | $l$ | $u$ |
| Rhino | yes | 0.04 | 0.11 |
| Rhino | no | 0.1 | 0.15 |
| Donkey | yes | 0.22 | 0.27 |
| Donkey | no | 0.09 | 0.16 |
| Elephant | yes | 0.05 | 0.13 |
| Elephant | no | 0.21 | 0.26 |
| mayor: | Donkey{senate, legalization} | | |

$\Longrightarrow$

| $\omega$: | $\pi_{\{\mathsf{senate}\}}(S_2)$ | |
|---|---|---|
| date: | October 23 | |
| gender: | male | |
| respondents: | 238, {senate} | |
| respondents: | 195, {legalization} | |
| overlap: | 184 | |
| senate | $l$ | $u$ |
| Rhino | 0.04 | 0.11 |
| Rhino | 0.1 | 0.15 |
| Donkey | 0.22 | 0.27 |
| Donkey | 0.09 | 0.16 |
| Elephant | 0.05 | 0.13 |
| Elephant | 0.21 | 0.26 |
| mayor: | Donkey{senate, legalization} | |

$\Longrightarrow$

| $\omega$: | $\pi_{\{\mathsf{senate}\}}(S_2)$ | |
|---|---|---|
| date: | October 23 | |
| gender: | male | |
| respondents: | 238, {senate} | |
| overlap: | 184 | |
| senate | $l$ | $u$ |
| Rhino | 0.14 | 0.26 |
| Donkey | 0.31 | 0.43 |
| Elephant | 0.26 | 0.39 |
| mayor: | Donkey{senate} | |

$\Longrightarrow$

| $\omega$: | $\pi_{\{\mathsf{senate}\}}(S_2)$ | |
|---|---|---|
| date: | October 23 | |
| gender: | male | |
| respondents: | 238, {senate} | |
| overlap: | 184 | |
| senate | $l$ | $u$ |
| Rhino | 0.18 | 0.26 |
| Donkey | 0.35 | 0.43 |
| Elephant | 0.31 | 0.39 |
| mayor: | Donkey{senate} | |

Figure 10: Projection on the participating random variables.

## 5.3 Conditionalization

Conditionalization was first considered as an operation of a relational algebra related to a probabilistic data model by Dey and Sarkar [13]; Classical relational algebra has no prototype of it. Intuitively, conditionalization is the operation of computing a conditional probability distribution, given a joint probability distribution. To simplify the definition below, we will employ the following notation. Let $V = \{v_1, \ldots, v_n\}$ be a set of random variables and let $v \in V$ and $V' = V - \{v\}$. Let $I : dom(V) \to [0,1]$ be a p-interpretation. Let $X = \{x_1, \ldots x_m\} \subset dom(v)$ and $\bar{y} \in dom(V')$. Then $I[X](\bar{y})$ denotes the following sum: $I[X](\bar{y}) = \sum_{i=1}^{m} I(\bar{y}, x_i)$. With this notation in mind, we define conditionalization as follows.

**Definition 29** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ be an ESPO, $|V| > 1$, $v \in V$ and $c : v = \{x_1, \ldots, x_m\}$ be a conditional selection condition. Then, the result of* ***conditionalization of*** *$S$* ***on*** *$c$, denoted $\mu_c(S)$ is the ESPO $S' = \langle T^+, V', P', C^{+'}, \omega'\rangle$, where*

- $V' = V - \{v\}$. *Without loss of generality, we will assume further that* $V = \{v_1, \ldots, v_n\}$, $v = v_n$ *and therefore* $V' = \{v_1, \ldots, v_{n-1}\}$.
- $C^{+'} = C^+ \cup \{(v, X, V')\}$, *where* $X = \{x_1, \ldots, x_m\}$.
- $P' : dom(V') \to$ ***C[0,1]*** *is defined as* [3]

$$P'(\bar{y}) = \left[ \min_{I \models P} \left( \frac{I_X(\bar{y})}{\sum_{y' \in dom(V')} I_X(\bar{y'})} \right), \max_{I \models P} \left( \frac{I_X(\bar{y})}{\sum_{y' \in dom(V')} I_X(\bar{y'})} \right) \right].$$

- $\omega' = \mu_c(\omega)$.

From the definition above, it follows that in order to compute the result of conditionalization of an ESPO (in particular, in order to compute the resulting probability distribution) a number of non-linear optimization problems have to be solved. As it turns out, the new probability distribution can be computed directly (i.e., both minimization and maximization problems that need to be solved have analytical solutions).

**Theorem 5** *Let* $S = \langle T^+, V, P, C^+, \omega \rangle$ *be an ESPO,* $c : v = \{x_1, \ldots, x_m\}$ *be a conditional selection condition and* $v \in V$*. Let* $V' = V - \{v\}$, $X = \{x_1, \ldots, x_m\}$ *and* $\bar{y} \in dom(V')$*. The result of the conditionalization is denoted* $S' = \mu_c(S) = \langle T^+, V', P', C^{+'}, \omega' \rangle$*. If we define* $l[X]_{\bar{y}}$ *and* $u[X]_{\bar{y}}$ *as follows:*

$$l[X]_{\bar{y}} = \max \left( \sum_{x \in X} l_{(\bar{y}, x)} \; ; \; 1 - \sum_{\bar{y'} \neq \bar{y} \textbf{ or } x' \notin X} u_{(\bar{y'}, x')} \right),$$

$$u[X]_{\bar{y}} = \min \left( 1 - \sum_{\bar{y'} \neq \bar{y} \textbf{ or } x' \notin X} l_{(\bar{y'}, x')} \; ; \; \sum_{x \in X} u_{(\bar{y}, x)} \right),$$

*then the following expression correctly computes the lower and upper bounds of the conditional probability distribution for the resulting ESPO object.*

$$P'(\bar{y}) = \left[ \frac{l[X]_{\bar{y}}}{\min \left( 1 - \sum_{x' \notin X} l_{(\bar{y'}, x')} \, , \, \sum_{\bar{y^*} \neq \bar{y} \, , \, x \in X} u_{(\bar{y^*}, x)} + l[X]_{\bar{y}} \right)}, \right.$$
$$\left. \frac{u[X]_{\bar{y}}}{\max \left( \sum_{\bar{y^*} \neq \bar{y} \, , \, x \in X} l_{(\bar{y^*}, x)} + u[X]_{\bar{y}} \, , \, 1 - \sum_{x' \notin X} u_{(\bar{y'}, x')} \right)} \right].$$

The proof of this theorem can be found in [9]. The following example will illustrate how the conditionalization operation works.

**Example 20** *Consider the ESPO* $S_2$ *in Figure 5. In this example, we illustrate the process of computing the conditionalization* $\mu_{\{\mathsf{legalization=yes}\}}(S_2)$*, as shown in Figure 11. First we collapse all the rows that do not satisfy the condition* legalization = yes *into one row. Next, we do a tightening operation on the new probability distribution. Then what we need to do is a normalization operation, which means that we must*

[3] We note, however, that Jaffray [16] has shown that conditioning interval probabilities is a dicey matter: the set of point probability distributions represented by $P'(\bar{y})$ *will* contain distributions $I'$ which do not correspond to any $I$ in $P$.

*find the minimum and maximum values of the expressions of the form* $\frac{\mathsf{I(w,yes)}}{\mathsf{I(Rhino,yes)+I(Donkey,yes)+I(Elephant,yes)}}$ *for* $w \in \{\mathsf{Rhino}, \mathsf{Donkey}, \mathsf{Elephant}\}$ *over all p-interpretations* $I \models P$.

*Let us determine the lower bound for* $\mathsf{x} = \mathsf{Rhino}$. *Consider the following function* $f$ *of three variables:* $f(x,y,z) = \frac{x}{x+y+z}$. *For positive* $x$, $y$ *and* $z$, *we could rewrite the function as* $f(x,y,z) = \frac{1}{1+\frac{y+z}{x}}$ *So, in order to minimize* $f$ *we need to minimize x and maximize y+z. In this case, we need to minimize* $\mathsf{I(Rhino,yes)}$ *and maximize* $\mathsf{I(Donkey,yes)} + \mathsf{I(Elephant,yes)}$, *i.e.,* $\mathsf{I(Rhino,yes)} = 0.04$ *and* $\mathsf{I(Donkey,yes)} + \mathsf{I(Elephant,yes)} = \mathsf{min}(0.27 + 0.13, 1 - 0.04 - 0.49) = 0.4$. *Then the minimum value of* $\frac{\mathsf{I(Rhino,yes)}}{\mathsf{I(Rhino,yes)+I(Donkey,yes)+I(Elephant,yes)}}$ *is* $\frac{0.04}{0.04+0.4} = 0.09$.

| $\omega$: | $S_2$ | | |
|---|---|---|---|
| date: | October 23 | | |
| gender: | male | | |
| respondents: | 238, {senate} | | |
| respondents: | 195, {legalization} | | |
| overlap: | 184 | | |
| senate | legalization | $l$ | $u$ |
| Rhino | yes | 0.04 | 0.11 |
| Rhino | no | 0.1 | 0.15 |
| Donkey | yes | 0.22 | 0.27 |
| Donkey | no | 0.09 | 0.16 |
| Elephant | yes | 0.05 | 0.13 |
| Elephant | no | 0.21 | 0.26 |
| mayor: | Donkey{senate, legalization} | | |

$\Longrightarrow$

| $\omega$: | $\mu_{\{\mathsf{legalization=yes}\}}(S_2)$ | | |
|---|---|---|---|
| date: | October 23 | | |
| gender: | male | | |
| respondents: | 238, {senate} | | |
| respondents: | 195, {legalization} | | |
| overlap: | 184 | | |
| senate | legalization | $l$ | $u$ |
| Rhino | yes | 0.04 | 0.11 |
| Donkey | yes | 0.22 | 0.27 |
| Elephant | yes | 0.05 | 0.13 |
| / | no | 0.4 | 0.57 |
| mayor: | Donkey{senate, legalization} | | |

$\Longrightarrow$

| $\omega$: | $\mu_{\{\mathsf{legalization=yes}\}}(S_2)$ | | |
|---|---|---|---|
| date: | October 23 | | |
| gender: | male | | |
| respondents: | 238, {senate} | | |
| overlap: | 184 | | |
| senate | legalization | $l$ | $u$ |
| Rhino | yes | 0.04 | 0.11 |
| Donkey | yes | 0.22 | 0.27 |
| Elephant | yes | 0.05 | 0.13 |
| / | no | 0.49 | 0.57 |
| mayor: | Donkey{senate, legalization} | | |

$\Longrightarrow$

| $\omega$: | $\mu_{\{\mathsf{legalization=yes}\}}(S_2)$ | |
|---|---|---|
| date: | October 23 | |
| gender: | male | |
| respondents: | 238, {senate} | |
| overlap: | 184 | |
| senate | $l$ | $u$ |
| Rhino | 0.09 | 0.36 |
| Donkey | 0.48 | 0.53 |
| Elephant | 0.12 | 0.30 |
| mayor: | Donkey{senate} | |
| legalization: | yes{senate} | |

$\Longrightarrow$

| $\omega$: | $\mu_{\{\mathsf{legalization=yes}\}}(S_2)$ | |
|---|---|---|
| date: | October 23 | |
| gender: | male | |
| respondents: | 238, {senate} | |
| overlap: | 184 | |
| senate | $l$ | $u$ |
| Rhino | 0.17 | 0.36 |
| Donkey | 0.48 | 0.53 |
| Elephant | 0.12 | 0.30 |
| mayor: | Donkey{senate} | |
| legalization: | yes{senate} | |

Figure 11: Conditionalization operation.

*Similarly, we can determine the upper bound for* $\mathsf{x} = \mathsf{Rhino}$. *We need to maximize* $\mathsf{I(Rhino,yes)}$ *and minimize* $\mathsf{I(Donkey,yes)} + \mathsf{I(Elephant,yes)}$, *i.e.,* $\mathsf{I(Rhino,yes)} = 0.11$ *and* $\mathsf{I(Donkey,yes)} + \mathsf{I(Elephant,yes)} = \mathsf{max}(0.22 + 0.05, 1 - 0.11 - 0.57) = 0.32$. *Then the maximum value of* $\frac{\mathsf{I(Rhino,yes)}}{\mathsf{I(Rhino,yes)+I(Donkey,yes)+I(Elephant,yes)}}$ *is* $\frac{0.11}{0.11+0.32} = 0.36$. *We can apply similar operations for* $\mathsf{x} = \mathsf{Donkey}$ *and* $\mathsf{x} = \mathsf{Elephant}$. *After that, the tightening operation is performed to find the true intervals. Finally, we exclude* ***respondents:195*** *from the context as it is associated with* ***legalization*** *variable and add* ***legaliza-***

*tion=yes to the conditionals. The resulting ESPO is shown in the bottom of Figure 11.*

### 5.4 Cartesian Product and Join

The Cartesian product of two ESPOs can be viewed as the joint probability distribution of the random variables from both objects. As only point probabilities were used in [10], we made there an assumption of independence between the random variables in the SPOs being combined. As probability distribution functions considered here are interval, this restriction will be removed. Also, the use of extended context and extended conditionals in ESPOs will allow us to make *Cartesian product compatibility* much less restrictive.

#### 5.4.1 Probabilistic Conjunctions

Probabilistic conjunctions are interval functions (operations) that are used to compute the probability of a conjunction of two events given the probabilities of individual events. Typically, each probabilistic conjunction operation would have an *underlying assumption* about the relationship between the events involved, such as *independence*, *ignorance*, *positive* or *negative correlation*. Probabilistic conjunctions had been introduced by Lakshmanan et al. [18], where they have also been used in defining the Cartesian product operation. Our definitions are borrowed from [18] and [11].

First, recall the standard "truth-ordering" on intervals: 1. $[L_1, U_1] \leq [L_2, U_2]$ iff $(L_1 \leq L_2 \wedge U_1 \leq U_2)$. 2. $[L_1, U_1] \geq [L_2, U_2]$ iff $(L_1 \geq L_2 \wedge U_1 \geq U_2)$.

**Definition 30 (probabilistic conjunction/disjunction strategy)** *A probabilistic conjunction $\otimes$ is a binary operation $\otimes$ : C[0,1] $\times$ C[0,1] $\rightarrow$ C[0,1] that obeys the following postulates:*

| Postulates | |
|---|---|
| 1. Commutativity | $([l_1, l_1] \otimes [l_2, u_2]) = ([l_2, u_2] \otimes [l_1, u_1])$ |
| 2. Associativity | $(([l_1, u_1] \otimes [l_2, u_2]) \otimes [l_3, u_3]) = ([l_1, u_1] \otimes ([l_2, u_2]) \otimes [l_3, u_3]))$ |
| 3. Monotonicity | $([l_1, u_1] \otimes [l_2, u_2]) \leq ([l_1, u_1] \otimes [l_3, u_3])$ *if* $[l_2, u_2] \leq [l_3, u_3]$ |
| 4. Bottomline | $([l_1, u_1] \otimes [l_2, u_2]) \leq [\min(l_1, l_2), \min(u_1, u_2)]$ |
| 5. Identity | $([l_1, u_1] \otimes [1, 1]) = [l_1, u_1]$ |
| 6. Annihilator | $([l_1, u_1] \otimes [0, 0]) = [0, 0]$ |
| 7. Ignorance | $([l_1, u_1] \otimes [l_2, u_2]) \subseteq [\max(0, L_1 + L_2 - 1), \min(U_1, U_2)]$ |

The following are some sample probabilistic conjunction operations ([11, 18]).

| Probabilistic Conjunctions | |
|---|---|
| Ignorance | $([l_1, u_1] \otimes_{ig} [l_2, u_2]) = [\max(0, l_1 + l_2 - 1), \min(u_1, u_2)]$ |
| Positive Correlation | $([l_1, u_1] \otimes_{pc} [l_2, u_2]) = [\min(l_1, u_2), \min(l_1, u_2)]$ |
| Negative Correlation | $([l_1, u_1] \otimes_{nc} [l_2, u_2]) = [\max(0, l_1 + l_2 - 1), \max(0, u_1 + u_2 - 1)]$ |
| Independence | $([l_1, u_1] \otimes_{in} [l_2, u_2]) = [l_1 \cdot l_2, u_1 \cdot u_2]$ |

#### 5.4.2 Cartesian Product

As different probabilistic conjunction operations compute the probabilities of conjunction of two events in different ways, there is no unique Cartesian product operation. Rather, for each probabilistic conjunction $\otimes_\alpha$ we define a Cartesian product operation $\otimes_\alpha$.

**Definition 31** *Let $S = \langle T^+, V, P, C^+, \omega \rangle$ and $S' = \langle T^{+'}, V', P', C^{+'}, \omega \rangle$ be two ESPOs. Let $V = \{v_1, \ldots, v_n\}$, $V' = \{v'_1, \ldots, v'_m\}$, $U = \{u \in \mathcal{V} | (u, X, V^*) \in C^+\}$, $U' = \{u' \in \mathcal{V} | (u', X', V^*) \in C^{+'}\}$. $S$ and $S'$ are* ***Cartesian product-compatible*** *iff (i) $V \cap V' = \emptyset$; (ii) $U \cap V' = \emptyset$, and (iii) $V \cap U' = \emptyset$.*

*Cartesian product compatibility* of two ESPOs means that the joint probability distribution of the random variables from both objects is meaningful. In particular, we require the sets of participating random variables to be disjoint (leaving the other case to be handled by the join operation). We also want the set of random variables found in the conditionals of one ESPO to be disjoint from the participating variables of the other. Thus, for example, Cartesian product of the probability distribution of mayor votes for respondents who will vote Donkey for senate cannot be combined with the probability distribution of senate votes. We can now define Cartesian product.

**Definition 32** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ and $S' = \langle T^{+'}, V', P', C^{+'}, \omega'\rangle$ be two Cartesian-product compatible ESPOs. Let $V = \{v_1, \ldots, v_n\}$, $V' = \{v'_1, \ldots, v'_m\}$, $U = \{u \in \mathcal{V} | (u, X, V^*) \in C^+\}$, $U' = \{u' \in \mathcal{V} | (u', X', V^*) \in C^{+'}\}$. Let $\otimes_\alpha$ be some* probabilistic conjunction*. The Cartesian product of S and $S'$ under probabilistic conjunction $\otimes_\alpha$, denoted $S \times_\alpha S'$, is defined as $S \times_\alpha S' = S'' = \langle T^{+''}, V'', P'', C^{+''}, \omega''\rangle$, where*

- $V'' = V \cup V'$;
- $T^{+''} = \{(A, a, V^*) | (A, a, V^*) \in T^+$ and no $(A, a, V_*) \in T^{+'}$ **or** $(A, a, V^*) \in T^{+'}$ and no $(A, a, V_*) \in T^+$ **or** $(A, a, V^*_1) \in T^+$ and $(A, a, V^*_2) \in T^{+'}$ and $V^* = V^*_1 \cup V^*_2\}$;
- $C^{+''} = \{(u, X, V^*) | (u, X, V^*) \in C^+$ and no $(u, X, V_*) \in T^{+'}$ **or** $(u, X, V^*) \in T^{+'}$ and no $(u, X, V_*) \in T^+$ **or** $(u, X, V^*_1) \in T^+$ and $(u, X, V^*_2) \in T^{+'}$ and $V^* = V^*_1 \cup V^*_2\}$;
- *$P'' : dom(V'') \to$ C[0,1] is defined as follows. Let $\bar{x} \in dom(V)$, $\bar{x'} \in domV'$ (hence $(\bar{x}, \bar{x'}) \in dom(V'')$). Then, $P''((\bar{x}, \bar{x'})) = P(\bar{x}) \otimes_\alpha P'(\bar{x'})$.*
- $\omega'' =$ "$\omega \times_\alpha \omega'$".

In Cartesian products the context and the conditionals of the two initial ESPOs are united; if a particular context record or a conditional appears in both ESPOs then their association lists are merged. The new set of participating variables is the union of the two original sets. Finally, the probability interval for each instance (row) of the new probability table is computed by applying the probabilistic conjunction operation to the appropriate rows of the two original tables.

### 5.4.3 Join

Join in ESP-Algebra is similar to Cartesian product in that it computes the joint probability distribution of the input ESPOs. The difference is that join is applicable to the ESPOs that have *common participating random variables*. Let $S = \langle T^+, V, P, C^+\omega\rangle$ and $S' = \langle T^{+'}, V', P', C^{+'}\omega\rangle$, and let $V^* = V \cap V' \neq \emptyset$ and participating random variables of $S$ are not conditioned in $S'$ and vice versa. If these conditions are satisfied, we call $S$ and $S'$ *join-compatible*.

**Definition 33** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ and $S' = \langle T^{+'}, V', P', C^{+'}, \omega'\rangle$ be two ESPOs. Let $V = \{v_1, \ldots, v_n\}$, $V' = \{v'_1, \ldots, v'_m\}$, $U = \{u \in \mathcal{V} | (u, X, V^*) \in C^+\}$, $U' = \{u' \in \mathcal{V} | (u', X', V^*) \in C^{+'}\}$. $S$ and $S'$ are **join-compatible** iff (i)$V \cap V' \neq \emptyset$; (ii) $U \cap V' = \emptyset$, and (iii) $V \cap U' = \emptyset$.*

Consider three value vectors $\bar{x} \in dom(V - V^*)$, $\bar{y} \in dom(V^*)$ and $\bar{z} \in dom(V' - V^*)$. The join of $S$ and $S'$ is the joint probability distribution $P''(\bar{x}, \bar{y}, \bar{z})$ of $V$ and $V'$, or, more specifically, of $V - V^*$, $V^*$ and $V' - V^*$. To construct this joint distribution, we recall from probability theory that under assumption $\alpha$ about the relationship between the random variables in $V$ and $V'$ and independence between variables in $V - V^*$ and in $V' - V^*$, we have $p(\bar{x}, \bar{y}, \bar{z}) = p(\bar{x}, \bar{y}) \otimes_\alpha p(\bar{z}|\bar{y})$ and, symmetrically, $p(\bar{x}, \bar{y}, \bar{z}) = p(\bar{x}|\bar{y}) \otimes_\alpha p(\bar{y}, \bar{z})$.

$p(\bar{x},\bar{y})$ is stored in $P$, the probability table of $S$. $p(\bar{z}|\bar{y})$ is the conditional probability that can be found by conditioning $p(\bar{y},\bar{z})$ (stored in $P'$) on $\bar{y}$. The second equality can be exploited in the same manner.

This gives rise to two families of join operations, left join ($\ltimes_\alpha$) and right join ($\rtimes_\alpha$) defined as follows.

**Definition 34** *Let $S = \langle T^+, V, P, C^+, \omega\rangle$ and $S' = \langle T^{+'}, V', P', C^{+'}, \omega'\rangle$ be two join-compatible ESPOs. Let $V^* = V \cap V' \neq \emptyset$ and $V_1 = V - V^*$ and $V_1' = V' - V^*$. We define the operations of left join of $S$ and $S'$, denoted $S \ltimes_\alpha S'$ and right join of $S$ and $S'$, denoted $S \rtimes_\alpha S'$ under assumption $\alpha$ as follows:*

$$S \ltimes S' = S'' = \langle T^{+''}, V'', P'', C^{+''}, \omega''\rangle;$$
$$S \rtimes S' = S''' = \langle T^{+''}, V'', P''', C^{+''}, \omega'''\rangle,$$

*where*

- $T^{+''} = T^+ \cup T^{+'}$;
- $V'' = V_1 \cup V^* \cup V_1'$;
- $P'', P''' : dom(V'') \longrightarrow$ *C[0,1].*
  *For all $\bar{w} \in dom(V'')$; $\bar{w} = (\bar{x},\bar{y},\bar{z})$; $\bar{x} \in dom(V_1)$, $\bar{y} \in dom(V^*)$, $\bar{z} \in dom(V_1')$:*
  *let $S_{\bar{y}} = \mu_{V^*=\bar{y}}(S) = \langle T^+, V - V^*, P_{\bar{y}}, C^+_{\bar{y}}\rangle$ and $S'_{\bar{y}} = \mu_{V^*=\bar{y}}(S') = \langle T^{+'}, V' - V^*, P'_{\bar{y}}, C^{+'}_{\bar{y}}\rangle$.*

$$P''(\bar{w}) = P_{\bar{y}}(\bar{x}) \otimes_\alpha P'(\bar{y},\bar{z});$$
$$P'''(\bar{w}) = P((\bar{x},\bar{y})) \otimes_\alpha P'_{\bar{y}}(\bar{z}).$$

- $C^{+''} = C^+ = C^{+'}$.
- $\omega'' = \omega \ltimes_\alpha \omega'$; $\omega''' = \omega \rtimes_\alpha \omega'$.

**Example 21** *Consider the two ESPOs $S_2$ and $S_3$ in Figure 5. They are joint probability distributions for (senate, legalization) and (park, legalization), respectively. However, in some circumstances we may want to combine these two ESPOs and obtain the joint probability distribution for all of the three random variables. We may apply a join operation to these ESPOs since they are join-compatible according the definition. We'll illustrate how to obtain the left join under the assumption of independence: $S_2 \ltimes_{ind} S_3$, as follows.*

*Join operation combines three operations: conditionalization, selection and Cartesian product. First, we need to calculate the results for conditionalization of the left operand (i.e. $S_2$) on the set of common variables (in this case, legalization), as shown in the top right part of Figure 12. Second, we do selections on probability table for all the possible values of the common variables, namely, $\sigma_{legalization=yes}(S_3)$ and $\sigma_{legalization=no}(S_3)$. The resulting ESPOs are shown in the bottom left part. Third, Cartesian product operations on corresponding ESPOs [4] are applied based on the values of the common variables, namely, $\mu_{legalization=yes}(S_2) \times_\alpha \sigma_{legalization=yes}(S_3)$ and $\mu_{legalization=no}(S_2) \times_\alpha \sigma_{legalization=no}(S_3)$. In this particular example, we assume that the random variables in the two ESPOs are independent when we apply probability conjunctions. Finally, we union all the resulting ESPOs and apply a tightening operation on it. The final result is shown in the bottom right part of the figure.*

[4] Prior to applying Cartesian product operation, we project legalization = yes and legalization = no out of the conditionals.

$\omega$: $S_2$
date: October 23
gender: male

| senate | legalization | $l$ | $u$ |
|---|---|---|---|
| Rhino | yes | 0.04 | 0.11 |
| Rhino | no | 0.1 | 0.15 |
| Donkey | yes | 0.22 | 0.27 |
| Donkey | no | 0.09 | 0.16 |
| Elephant | yes | 0.05 | 0.13 |
| Elephant | no | 0.21 | 0.26 |

mayor: Donkey

$\omega$: $\mu_{\text{legalization=yes}}(S_2)$
date: October 23
gender: male

| senate | $l$ | $u$ |
|---|---|---|
| Rhino | 0.17 | 0.36 |
| Donkey | 0.48 | 0.53 |
| Elephant | 0.12 | 0.30 |

mayor: Donkey
legalization: yes

$\omega$: $\mu_{\text{legalization=no}}(S_2)$
date: October 23
gender: male

| senate | $l$ | $u$ |
|---|---|---|
| Rhino | 0.17 | 0.36 |
| Donkey | 0.48 | 0.53 |
| Elephant | 0.12 | 0.30 |

mayor: Donkey
legalization: no

$\omega$ $S_3$
locality: Sunny Hill
date: October 26

| park | legalization | $l$ | $u$ |
|---|---|---|---|
| yes | yes | 0.56 | 0.62 |
| yes | no | 0.14 | 0.2 |
| no | yes | 0.21 | 0.25 |
| no | no | 0.03 | 0.07 |

major: Donkey

$\omega$ $\sigma_{\text{legalization=yes}}(S_3)$
locality: Sunny Hill
date: October 26

| park | legalization | $l$ | $u$ |
|---|---|---|---|
| yes | yes | 0.56 | 0.62 |
| no | yes | 0.21 | 0.25 |

major: Donkey

$\omega$ $\sigma_{\text{legalization=no}}(S_3)$
locality: Sunny Hill
date: October 26

| park | legalization | $l$ | $u$ |
|---|---|---|---|
| yes | no | 0.14 | 0.2 |
| no | no | 0.03 | 0.07 |

major: Donkey

$\omega$: $S_2 \ltimes S_3$
locality: Sunny Hill
date: October 26
gender: male

| senate | park | legalization | l | u |
|---|---|---|---|---|
| Rhino | yes | yes | 0.09 | 0.22 |
| Rhino | yes | no | 0.03 | 0.06 |
| Rhino | no | yes | 0.04 | 0.09 |
| Rhino | no | no | 0.01 | 0.02 |
| Donkey | yes | yes | 0.27 | 0.33 |
| Donkey | yes | no | 0.03 | 0.07 |
| Donkey | no | yes | 0.11 | 0.13 |
| Donkey | no | no | 0.01 | 0.02 |
| Elephant | yes | yes | 0.07 | 0.19 |
| Elephant | yes | no | 0.06 | 0.11 |
| Elephant | no | yes | 0.03 | 0.07 |
| Elephant | no | no | 0.01 | 0.04 |

major: Donkey

Figure 12: Join operation (left join) in ESP-Algebra

# 6 Related Work

This work builds on the work of many people in two fields: imprecise probabilities and probabilistic databases. The overlap between these two fields is still small, so we address them separately. Databases that handle imprecise probabilities are surveyed in Section 6.2.

## 6.1 Interval Probabilities

Imprecise probabilities have attracted the attention of researchers for quite a while now, as documented by the Imprecise Probability Project [21]. Walley's seminal work [20] made the case for interval probabilities as the means of representing uncertainty. In his book, Walley talks about the computation of conditional probabilities of events. As mentioned in Section 1, his semantics is quite different from ours, as Walley constructs his theory of imprecise probabilities based on gambles and betting, expressed as lower and upper previsions on the sets of events. Conditional probabilities are also specified via gambles by means of *conditional previsions*. A similar approach to Walley's is found in the work of Biazzo, Gilio, et al. [2, 3] where they extend the theory of imprecise probabilities to incorporate logical inference and default reasoning.

Walley [20] calls consistency and tightness properties "avoiding sure loss", and "coherence", respectively. Biazzo and Gilio [2] also use the term "g-coherence" as a synonym for "avoiding sure loss". The terminology that we have adopted originated from the work of Dekhtyar, Ross and Subrahmanian on a specialized semantics for probability distributions used in their Temporal Probabilistic Database model [11]. However, the semantics presented here is a significant generalization of the their semantics. The possible world semantics for interval probabilities also occurs in Givan, Leach and Dean's discussion of Bounded Parameter Markov Decision Processes [14].

De Campos, Huete and Morel [8] studied probability intervals as a tool to represent uncertain information. They gave similar definitions as we do for consistency and tightness, which they call reachability. They developed a calculus for probability intervals, including combination, marginalization and conditioning. They also explored the relationship of their formalism with other uncertain theories, such as lower and upper probabilities. When they defined their conditioning operation, however, they switched back and applied lower and upper probabilities to uncertain information instead of probability intervals, and gave a definition of conditioning operation for bidimensional probability intervals. Ours extends their definition.

A more direct approach to introducing interval probabilities is found in the work of Weichselberger [22] who extends the Kolmogorov axioms of probability theory to the case of interval probabilities. Building on Kolmogorov probability theory, the interval probability semantics is defined for a $\sigma$-algebra of random events. Weichselberger defines two types of interval probability distributions over this $\sigma$-algebra: R-Probabilities, similar to our consistent interval pdfs and F-Probabilities, similar to our tight interval pdfs. In his semantics an event is specified as a Boolean combination of atomic events from some set $\Omega$. Each event partitions the set of possible worlds into two sets: those in which the event has occurred and those in which it has not. A lower bound on the probability that an event has occurred is immediately an upper bound on the probability that it has not occurred. Thus, for F-probabilities, Weichselberger's analogs of our tight p-interpretations, lower bounds uniquely determine upper bounds.

Weichselberger completes his theory with two definitions of conditional probability: "intuitive" and "formal". His "intuitive" definition semantically matches our Definition 29. On the other hand, the "formal" definition specifies the probability interval for $P(A|B)$ as $[\frac{\min(P(AB))}{\min P(B)}, \frac{\max(P(AB))}{\max P(B)}]$, which is somewhat different from our Theorem 5. There, to determine the lower bound we minimize the numerator and try to *maximize* the denominator. Similarly, for the upper bound, we maximize the numerator and *minimize* the denominator.

In our semantics, atomic events have the form "random variable $X_1$ takes value $a_1$ and random variable $X_2$ takes value $a_2$ and ... and random variable $X_m$ takes value $a_m$." The negation of such an event is the disjunction of all other atomic events that complete the joint probability distribution of random variables $X_1, \ldots, X_m$. Our interval pdfs specify only the probability intervals for such atomic events, without explicitly propagating them onto the negations. This means that even for tight interval pdfs, both upper and lower bounds are necessary in all but marginal cases, as illustrated in Figure 13.

Interval probability distributions of discrete random variables generate a set of linear constraints on the acceptable probability values for individual instances. This set of linear constraints, however, is quite simple. It consists of constraints specifying that the probabilities of individual instances must fall between the given lower and upper bounds and a constraint that specifies that the sum of all probabilities must be equal to 1. It is possible, however, to study more complex collections of constraints on possible worlds. Significant work in this area has been done by Cano and Moral [6].

| X | $l$ | $u$ |
|---|---|---|
| a | 0.3 | 0.4 |
| b | 0.4 | 0.5 |
| c | 0.2 | 0.3 |

| X | $l$ | $u$ |
|---|---|---|
| a | 0.3 | 0.35 |
| b | 0.4 | 0.45 |
| c | 0.2 | 0.27 |

Figure 13: Lower bounds do not uniquely define upper bounds for tight interval pdfs.

### 6.2 Probabilistic Databases

Cavallo and Pittarelli [7] were among the first to address the problem of storing and querying probabilistic information in a database. Their probabilistic relations resemble a single probability table from our ESPO. Their data model requires that the probabilities for all the tuples in a relation add up to exactly 1. As a result, unlike ours, their model requires a separate relation for each object. Barbara, Garcia-Molina and Porter in [1] proposed a new approach to managing probabilistic information. In their model, certain attributes in a relation could be designated as *stochastic* and (possibly joint) probability distributions could be associated with these attributes. The analogue of their non-stochastic attributes in our framework is context, while stochastic attributes are represented as participating random variables. The model of Barbara et al. was relational, and hence, the probability distributions stored in a single probabilistic relation had to be of the same structure.

Dey and Sarkar [13] introduced a 1NF probabilistic relational model and relational algebra. A tuple in their model is analogous to a single row of a probability table in ours, and their probabilistic relation could contain multiple probability distributions. Both [1] and [13] used point probabilities and assumed that all events/random variables in their models were independent. Lakshmanan et al. introduced ProbView [18], a probabilistic database management system. In ProbView, probability distributions were interval, and the assumption of independence of events had been replaced with the introductions of probabilistic conjunctions (and disjunctions), implementing different assumptions about the relationships between the events. Based on ProbView model, Dekhtyar, Ross and Subrahmanian developed Probabilistic Temporal Databases (TP-Databases) [11], a special-purpose probabilistic database model for managing temporal uncertainty. In this work, the semantics of interval probability distributions similar to the one used in ESPO model had been introduced, and the concept of *tightness* appeared for the first time in database literature.

Dey and Sarkar [13] first introduced the conditionalization operation in a probabilistic database model. Dekhtyar, Goldsmith and Hawkes also use this operation in their Semistructured Probabilistic Algebra [10]. In both works, conditionalization is performed on *point probability distributions* of discrete random variables, and the operation itself is fairly straightforward for point probability. The conditionalization operation as a database operation for probability intervals was not included in data models until recently by Goldsmith, Dekhtyar and Zhao [15]. We note, however, that Jaffray [16] has shown that interval conditional probability estimates will not be perfect, and that the unfortunate consequence of this is that conditionalizing is not commutative: $P((A|B)|C) \neq P(A|(B|C))$ for many $A$, $B$, and $C$. Thus, a conditionalization operation is included into ESP-Algebra with the caveat that the user must take care in the use of and interpretation of the result.

There are two approaches to semistructured probabilistic data management that are closely related to ours: the ProTDB [23] and the PIXml [17] frameworks. In ProTDB [23], Nierman and Jagadish extended the XML data model by associating a probability to each element with modification of regular non-probabilistic DTDs. They provided two ways of modifying non-probabilistic DTDs, either by introducing to every element a probability attribute Prob to specify the probability of the particular element existing at the specific location of the XML document or by attaching a new sub-element called Dist to each element. One of the drawbacks of their model is that probabilities in an ancestor-descendant chain were related probabilistically, meaning that probabilities in the document were always conditional probability. All other probabilities were assumed to be independent. Hung et al [17] proposed a probabilistic interval XML data model with two types of semantics for uncertain data. The global interpretation is a distribution over an entire XML document, while the local interpretation specifies an object probability function for each non-leaf object. They also proposed a path expression-based query language to access stored information. This approach overcomes some drawbacks presented in [23]. The major difference between it and our work is that [17] is concerned with representation of uncertainty in the structure of XML documents. At the same time, ESPO model provides a semistructured data type for storing probability distributions found in different applications. Hung et al. use our conditionalization formulae for their computations of conditional probabilities. This makes the two approaches comparable: our SPO objects can be represented as their probabilistic XML. At the same time, we can represent their probabilistic XML documents as joint probability distributions, and thus embed them into ESPO model. At the same time, while ESPOs are representable in XML, our definitions of the model and ESP-Algebra do not rely on a specific representation.

# 7 Conclusions and Future Work

Extended Semistructured Probabilistic Objects and Extended Semistructured Probabilistic Algebra introduced here represent a flexible database framework for storing and managing diverse probabilistic information. While such operations as probabilistic table selection, projection and conditionalization have been defined via the underlying semantics (i.e., in terms of satisfying p-interpretations), we have been able to provide direct ways of computing the results of these operations in each case, which lead to clear and efficient algorithms.

We have implemented an SPO database management system SPDBMS on top of a RDBMS, and testing on each query algebra operation has been conducted. Currently we are working on implementing the query optimizer. Implementation of extension to the SPO database management system to handle probability intervals has been underway. In the near future, we will study data fusion and conflict resolution problems that arise in this framework.


### Acknowledgements

This paper is a significant extension of the work presented in [9, 15, 24]. We'd like to thank the anonymous reviewers of our papers, whose suggestions improved this paper.


# References


[1] D. Barbara, H. Garcia-Molina and D. Porter. (1992) The Management of Probabilistic Data, *IEEE Trans. on Knowledge and Data Engineering*, Vol. 4, pp. 487–502.

[2] Veronica Biazzo, A. Gilio (2000) A generalization of the fundamental theorem of de Finetti for imprecise conditional probability assessments, *International Journal of Approximate Reasoning*, 24(2), pp. 251–272.

[3] V. Biazzo, A. Gilio, T. Lukasiewicz, G. Sanfilippo. (2001) Probabilistic Logic under Coherence, Model-Theoretic Probabilistic Logic, and Default Reasoning, *Proc. ECSQARU'2001, LNAI*, Vol. 2143, pp. 290–302

[4] G. Boole. (1854) *The Laws of Thought*, Macmillan, London.

[5] T. Bray, J. Paoli, C.M. Spreberg-McQueen. (Eds.) (1998) Extensible Markup Language (XML) 1.0, *World Wide Web Consortium Recommendation*, http://www.w3.org/TR/1998/REC-xml-19980210.

[6] A. Cano, S. Moral. (2000) Using probability trees to compute marginals with imprecise probabilities, *Universidad de Granada, Escuela Técnica Superior de Ingenieria Informática technical report, DECSAI-00-02-14*.

[7] R. Cavallo, M. Pittarelli. (1987) The Theory of Probabilistic Databases, *Proc. VLDB'87*, pp. 71-81.

[8] Luis M. de Campos, Juan F. Huete, Serafin Moral (1994) Probability Intervals: A Tool for Uncertain Reasoning, *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2(2), pp. 167–196, 1994.

[9] A. Dekhtyar, J. Goldsmith. (2002) Conditionalization for Interval Probabilities, *Proc. Workshop on Conditionals, Information, and Inference*, May, 2002; to appear in a Springer *Lecture Notes in Artificial Intelligence* volume based on the workshop.

[10] A. Dekhtyar, J. Goldsmith, S.R. Hawkes. (2001) Semistructured Probabilistic Databases, in *Proc. SSDBM'2001*.

[11] A. Dekhtyar, R. Ross, V.S. Subrahmanian. (2001) Temporal Probabilistic Databases, I: Algebra, *ACM Transactions on Database Systems*, vol 26, 1, pp. 41–95.

[12] A. Dekhtyar and V.S. Subrahmanian. (2000) Hybrid Probabilistic Logic Programs, *Journal of Logic Programming*, vol 43, 3, pp. 187–250.

[13] D. Dey and S. Sarkar. (1996) A Probabilistic Relational Model and Algebra, *ACM Transactions on Database Systems*, Vol. 21, 3, pp. 339–369.

[14] R. Givan, S. Leach, T. Dean. (2000) Bounded-Parameter Markov Decision Processes, *Artificial Intelligence*, Vol. 122, 1-2, pp. 71–109.

[15] J. Goldsmith, A. Dekhtyar, W. Zhao. (2003) Can Probabilistic Databases Help Elect Qualified Officials?, Proc. Florida AI Research Symposium, pp. 501–505.

[16] J.Y. Jaffray (1992) Bayesian Updating and Belief Functions. *IEEE Transactions on Systems, Man, and Cybernetics*, 22(5), pp. 1144–1152.

[17] E. Hung, L. Getoor, V.S. Subrahmanian. (2003) Probabilistic Interval XML, *Proc. International Conference on Database Theory*. pp. 361–377

[18] V.S. Lakshmanan, N. Leone, R. Ross and V.S. Subrahmanian. (1997) ProbView: A Flexible Probabilistic Database System. *ACM Transactions on Database Systems*, Vol. 22, No. 3, pp.419–469.

[19] R. Ramakrishnan, J Gehrke. (1999) *Database Management Systems*, 2nd Ed. McGraw-Hill.

[20] Walley, P. (1991). Statistical Reasoning with Imprecise Probabilities. Chapman and Hall, 1991.

[21] Gert de Cooman and Peter Walley *The Imprecise Probabilistic Project* URL: http://ippserv.rug.ac.be

[22] Weichselberger, K. (1999). The theory of interval-probability as a unifying concept for uncertainty. *Proc. 1st International Symp. on Imprecise Probabilities and Their Applications*.

[23] Andrew Nierman and H. V. Jagadish. (2002) ProTDB: Probabilistic Data in XML. *Proc. the 28th International VLDB Conference*. Hong Kong, China.

[24] W. Zhao, Alex Dekhtyar and Judy Goldsmith. (2003). Query Algebra for Interval Probabilities *Accepted to 14th International Conference on Database and Expert Systems Applications*.

# Proofs for ESPO Framework.

**Proof of Theorem 1**
*We prove the theorem of consistency.*

Let $P$ be an interval pdf over $V$ and let $dom(V) = \{\bar{x}_1, \ldots, \bar{x}_m\}$. Remember that we denote $P(\bar{x}_i)$ as $[l_i, u_i]$. Consider now two functions $f_l, f_u : dom(V) \to [0,1]$ such that $f_l(\bar{x}_i) = l_i$ and $f_u(\bar{x}_i) = u_i$ for all $1 \le i \le m$.
First we prove $P$ is consistent if $\sum_{i=1}^m l_i \le 1$ and $\sum_{i=1}^m u_i \ge 1$.

If $\sum_{i=1}^m l_i = 1$ then $f_l$ is a p-interpretation and $f_l \models P$. Therefore $P$ is consistent.

If $\sum_{i=1}^m u_i = 1$ then $f_u$ is a p-interpretation and $f_u \models P$ Therefore $P$ is consistent.

Consider now the case when $\sum_{i=1}^m l_i < 1$ and $\sum_{i=1}^m u_i > 1$. Let $\sum_{i=1}^m l_i = L$ and $\sum_{i=1}^m u_i = U$. We know that $L < 1 < U$.

Consider a function $I : dom(V) \to [0,1]$ such that

$$I(\bar{x}_i) = \frac{1-L}{U-L} u_i + \left(1 - \frac{1-L}{U-L}\right) l_i.$$

We now show that $I$ is a p-interpretation and $I \models P$. Let $\alpha = \frac{1-L}{U-L}$. As $L < 1 < U$, $0 < \alpha < 1$ and we can rewrite the definition of $I$ as $I(\bar{x}_i) = l_i + \alpha(u_i - l_i)$. Then $l_i \le I(\bar{x}_i) \le u_i$. Thus, if $I$ is a p-interpretation then $I \models P$.

To show that $I$ is a p-interpretation we need $\sum_{i=1}^m I(\bar{x}_i) = 1$. This can be demonstrated as follows:

$$\begin{aligned}\sum_{i=1}^m I(\bar{x}_i) &= \sum_{i=1}^m \left(\tfrac{1-L}{U-L} u_i + (1 - \tfrac{1-L}{U-L}) l_i\right)\\ &= \alpha \sum_{i=1}^m u_i + (1-\alpha) \sum_{i=1}^m l_i\\ &= \alpha U + (1-\alpha) L\\ &= \tfrac{1-L}{U-L} U + (1 - \tfrac{1-L}{U-L}) L\\ &= \tfrac{(1-L)U + (U-L-1+L)L}{U-L}\\ &= \tfrac{U-L}{U-L} = 1.\end{aligned}$$

To complete the proof, we prove that if $P$ is consistent then $\sum_{i=1}^m l_i \le 1$ and $\sum_{i=1}^m u_i \ge 1$.

If $P$ is consistent, then there exists a p-interpretation $I : dom(V) \to [0,1]$, such that $I \models P$. Then, for each $\bar{x}_i, 1 \le i \le m$, we have $l_i \le I(\bar{x}_i) \le u_i$. But then,
$\sum_{i=1}^m l_i \le \sum_{i=1}^m I(\bar{x}_i) \le \sum_{i=1}^m u_i$.

As $I$ is a p-interpretation, $\sum_{i=1}^m I(\bar{x}_i) = 1$ and we immediately get $\sum_{i=1}^m l_i \le 1$ and $\sum_{i=1}^m u_i \ge 1$. $\square$

**Proof of Theorem 2**
*Here we prove the theorem of tightening operation.*

Let $P'(\bar{x}_i) = [\max(l_i, 1 - \sum_{j=1}^m u_j + u_i), \min(u_i, 1 - \sum_{j=1}^m l_j + l_i)]$. We need to prove two statements: $P \equiv P'$ and $P'$ is tight.
First we prove $P \equiv P'$.
Notice that for all $1 \le i \le m$, $[\max(l_i, 1 - \sum_{j=1}^m u_j + u_i), \min(u_i, 1 - \sum_{j=1}^m l_j + l_i)] \subseteq [l_i, u_i]$.

Indeed, $l_i \le \max(l_i, 1 - \sum_{j=1}^m u_j + u_i)$ and $\min(u_i, 1 - \sum_{j=1}^m l_j + l_i)] \subseteq [l_i, u_i] \le u_i$. Now, because $P$ is consistent, $(\forall 1 \le i \le m)$, $l_i \le u_i$ and $\sum_{j=1}^m l_j \le 1 \le \sum_{j=1}^m u_j$. But then $1 - \sum_{j=1}^m u_j \le 0$ and hence $1 - \sum_{j=1}^m u_j + u_i \le u_i$, and therefore $\max(l_i, 1 - \sum_{j=1}^m u_j + u_i) \le u_i$.

Similarly, we obtain $l_i \le \min(u_i, 1 - \sum_{j=1}^m l_j + l_i)]$. Finally, for $1 \le j \le m$, $l_j \le u_j$, $\sum_{j=1}^m l_j - l_i \le \sum_{j=1}^m u_j - u_i$ and therefore $1 - \sum_{j=1}^m u_j + u_i \le 1 - \sum_{j=1}^m l_j + l_i$. Therefore,

$$l_i \le \max(l_i, 1 - \sum_{j=1}^m u_j + u_i) \le \min(u_i, 1 - \sum_{j=1}^m l_j + l_i) \le u_i.$$

This means that $(\forall I : dom(V) \to [0,1])(I \models P' \Rightarrow I \models P)$.

We now need to show the inverse: $(\forall I : dom(V) \to [0,1])(I \models P \Rightarrow I \models P')$.

Let $I$ be a p-interpretation over $V$ and let $I \models P$. Therefore, $(\forall 1 \leq i \leq m)(l_i \leq I(\bar{x}_i) \leq u_i)$. We need to show $\max(l_i, 1 - \sum_{j=1}^{m} u_j + u_i) \leq I(\bar{x}_i) \leq \min(u_i, 1 - \sum_{j=1}^{m} l_j + l_i)$.

We show $\max(l_i, 1 - \sum_{j=1}^{m} u_j + u_i) \leq I(\bar{x}_i)$. The other inequality can be proven similarly.

We know that $l_i \leq I(\bar{x}_i)$, so if $\max(l_i, 1 - \sum_{j=1}^{m} u_j + u_i) = l_i$ then the inequality holds.

Assume now that $\max(l_i, 1 - \sum_{j=1}^{m} u_j + u_i) = 1 - \sum_{j=1}^{m} u_j + u_i$. Since $l_i \geq 0$ and $1 - \sum_{j=1}^{m} u_j + u_i \geq 0$, we can have $\sum_{j=1}^{m} u_j - u_i \leq 1$.

Assume that the inequality does not hold, i.e., $I(\bar{x}_i) < 1 - \sum_{j=1}^{m} u_i + u_i$. We know that for all $1 \leq j \leq m$, $I(\bar{x}_j) \leq u_i$. Therefore $\sum_{j=1}^{m} I(\bar{x}_j) = \sum_{j=1, j \neq i}^{m} I(\bar{x}_j) + I(\bar{x}_i) \leq \sum_{j=1}^{m} u_j - u_i + I(\bar{x}_i) < \sum_{j=1}^{m} u_j - u_i + 1 - \sum_{j=1}^{m} u_j + u_i = 1$.

As $I$ is a p-interpretation, $\sum_{j=1}^{m} I(\bar{x}_j)$ must be equal to 1. This contradicts with $I(\bar{x}_i) \geq 1 - \sum_{j=1}^{m} u_j + u_i$.

Next we show that $P'$ is tight.

We show that for all $1 \leq i \leq m$, every point $a \in P'(\bar{x}_i)$ is *reachable*. By Proposition 1, it is sufficient to prove that the end points of the $P'(\bar{x}_i)$ interval are reachable.

Recall that $P'(\bar{x}_i) = [\max(l_i, 1-\sum_{j=1}^{m} u_j+u_i), \min(u_i, 1-\sum_{j=1}^{m} l_j+l_i)]$. We show that $\max(l_i, 1-\sum_{j=1}^{m} u_j+u_i)$ is reachable. Similar reasoning can be applied to show the reachability of the upper bound.

We show that there exists a p-interpretation $I$ such that $I \models P$ and $I(\bar{x}_i) = \max(l_i, 1 - \sum_{j=1}^{m} u_j + u_i)$. As we have shown that $P \equiv P'$, it follows that $I \models P'$.

First, suppose $\max(l_i, 1 - \sum_{j=1}^{m} u_j + u_i) = l_i$. Then $1 - l_i \leq \sum_{j=1}^{m} u_j - u_i$. Because $\sum_{j=1}^{m} l_j \leq 1$, we get $\sum_{j=1}^{m} l_j - l_i \leq 1 - l_i \leq \sum_{j=1}^{m} u_j - u_i$.

But then, by reasoning similar to that in the proof of Theorem 1, there exist numbers $a_1, \ldots, a_m$, such that $a_i = l_i$ and $(\forall 1 \leq j \leq m)(l_j \leq a_i \leq u_j)$ and $a_1 + \ldots + a_m = 1$. Let $I$ be a p-interpretation such that $I(\bar{x}_j) = a_j$ for all $1 \leq j \leq m$. Then $I(\bar{x}_i) = l_i$, $\sum_{j=1}^{m} I(\bar{x}_j) = 1$ and $I \models P$. Therefore $I \models P'$ and $l_i = \max(l_i, 1 - \sum_{j=1}^{m} u_j + u_i)$ is reachable.

Now suppose $\max(l_i, 1 - \sum_{j=1}^{m} u_j + u_i) = 1 - \sum_{j=1}^{m} u_j + u_i$.

Consider the function $I : dom(V) \rightarrow [0, 1]$, such that $I(\bar{x}_i) = 1 - \sum_{j=1}^{m} u_j + u_i$ and $I(\bar{x}_j) = u_j$ for all $1 \leq j \leq m, j \neq i$. If $I$ is a p-interpretation then $I \models P$ as $l_i \leq 1 - \sum_{j=1}^{m} u_j + u_i \leq u_i$ and $u_j \in [l_i, u_j]$. To prove that $I$ is a p-interpretation we must show that $\sum_{j=1}^{m} I(\bar{x}_j) = 1$.

Indeed, $\sum_{j=1}^{m} I(\bar{x}_j) = \sum_{j=1, j \neq i}^{m} I(\bar{x}_j) + I(\bar{x}_i) = \sum_{j=1}^{m} u_j - u_i + 1 - \sum_{j=1}^{m} u_j + u_i = 1$. This proves the reachability of $1 - \sum_{j=1}^{m} u_j + u_i = \max(l_i, 1 - \sum_{j=1}^{m} u_j + u_i)$, which, in turn proves the theorem. □

**Proof of Theorem 3**

*Here we prove that different selection operations commute.*

Let $c$ and $c'$ be two atomic selection conditions. There are 5 types of atomic selection conditions, namely *context*, *participation*, *conditional*, *table* and *probability*. Selection on *context*, *participation* or *conditional* will result in entire SPOs being selected, while selection on *table* or *probability* will select only parts of the relevant SPOs. We could partition the conditions into two groups,

- Group $I$, containing *context*, *extended context*, *participation*, *conditional* and *extended conditional* conditions, and
- Group $II$, containing *table* and *probability* conditions.

First we prove $\sigma_c(\sigma_{c'}(S)) = \sigma_{c'}(\sigma_c(S))$ for a single SPO $S$, and we consider here all the possible cases for each pair of condition groups.

**Case 1.** Both conditions $c$ and $c'$ are in Group $I$.

There are three possible combinations for whether each condition is satisfied:

a) $S$ satisfies $c$ but not $c'$, or
b) $S$ satisfies $c'$ but not $c$, or
c) $S$ satisfies both $c$ and $c'$, or
d) $S$ does not satisfy either $c$ or $c'$.

By the definition of selection on atomic selection conditions in Group $I$, we know selection on these conditions will result in the entire SPO being selected, or none of it.
For case a), since $S$ does not satisfy $c'$, $\sigma_{c'}(S)$ returns empty and subsequently $\sigma_c(\sigma_{c'}(S))$ will return empty. Since $\sigma_c(S)$ returns $S$, we see that $\sigma_{c'}(\sigma_c(S)) = \sigma_{c'}(S)$ will also return empty for the same reason. Thus, $\sigma_c(\sigma_{c'}(S)) = \sigma_{c'}(\sigma_c(S))$ holds for case (a). The same applies to case (b). Similarly, for case (d).
For case c), $\sigma_c(\sigma_{c'}(S)) = \sigma_c(S)$ returns $S$, and $\sigma_{c'}(\sigma_c(S)) = \sigma_{c'}(S)$ returns $S$ too. This proves that $\sigma_c(\sigma_{c'}(S)) = \sigma_{c'}(\sigma_c(S))$ holds for case (c).

So $\sigma_c(\sigma_{c'}(S)) = \sigma_{c'}(\sigma_c(S))$ holds for all the cases.

**Case 2.** Condition $c$ is in Group $I$ and condition $c'$ is in Group $II$.

There are only two possible combinations for whether each condition is satisfied, assuming that condition $c'$ is always partially satisfied:
a) $S$ does not satisfy $c$, or
b) $S$ satisfies $c$.

By the definition of selection on atomic selection conditions in both Group $I$ and Group $II$, we know selection on conditions in Group $I$ will result in the entire SPO being selected or not, while selection on conditions in Group $II$ will preserve all the context, participating random variables and conditionals in the original SPO, but produce only a part of the probability table.

Let $\sigma_{c'}(S) = S'$, where $S'$ has part of the probability table which satisfies the condition $c'$ and retains all the context, participating random variables and conditionals in $S$.

For case a), $\sigma_c(\sigma_{c'}(S)) = \sigma_c(S')$ will return empty since $S'$ does not satisfy the condition $c$ either. Since $\sigma_c(S)$ returns empty, subsequently $\sigma_{c'}(\sigma_c(S))$ will also return empty. This proves $\sigma_c(\sigma_{c'}(S)) = \sigma_{c'}(\sigma_c(S))$ for case (a).

For case (b), $\sigma_c(\sigma_{c'}(S)) = \sigma_c(S')$ will return $S'$ since $S'$ should satisfy the condition $c$ too. Since $\sigma_c(S)$ returns $S$, so $\sigma_{c'}(\sigma_c(S)) = \sigma_{c'}(S)$ will also return $S'$. This proves that $\sigma_c(\sigma_{c'}(S)) = \sigma_{c'}(\sigma_c(S))$ holds for case (b).

So $\sigma_c(\sigma_{c'}(S)) = \sigma_{c'}(\sigma_c(S))$ holds for both cases.

**Case 3.** Both $c$ and $c'$ are conditions in Group $II$.

First we prove $\sigma_c(\sigma_{c'}(S)) = \sigma_{c'}(\sigma_c(S))$ for a single SPO $S$. Assume that both conditions $c$ and $c'$ are partially satisfied by $S$. By the definition of selection on atomic selection conditions in Group $II$, we know selection on these conditions will result in part of the probability table and will preserve all the context, participating random variables and conditionals in the original SPO. In other words, all the components in the original SPO except the probability table will be preserved.

Let $S = \langle T, V, P, C\rangle$. Then $S' = \sigma_c(\sigma_{c'}(S)) = \langle T, V, P', C\rangle$ and $S'' = \sigma_{c'}(\sigma_c(S)) = \langle T, V, P'', C\rangle$ with $P' = \varrho_c(\varrho_{c'}(P)$, and $P'' = \varrho_{c'}(\varrho_c(P)$ where $\varrho$ is the relational selection operator. Since the relational selection operator $\varrho$ is commutative, $\varrho_c(\varrho_{c'}(P) = \varrho_{c'}(\varrho_c(P)$. Therefore we have $P' = P''$ or $S' = S''$. So $\sigma_c(\sigma_{c'}(S)) = \sigma_{c'}(\sigma_c(S))$ holds for this case.

Now let an SP-relation $\mathcal{S} = \cup_{S\in\mathcal{S}} S$. Since the union operator is commutative, $\sigma_{c'}(\sigma_c(\mathcal{S})) = \sigma_{c'}(\sigma_c(\cup_{S\in\mathcal{S}}(S))) = \cup_{S\in\mathcal{S}}(\sigma_{c'}(\sigma_c(S)))$, and $\sigma_c(\sigma_{c'}(\mathcal{S})) = \sigma_c(\sigma_{c'}(\cup_{S\in\mathcal{S}}(S))) = \cup_{S\in\mathcal{S}}(\sigma_c(\sigma_{c'}(S)))$.

So this proves $\sigma_c(\sigma_{c'}(\mathcal{S})) = \sigma_{c'}(\sigma_c(\mathcal{S}))$. □

**Proof of Theorem 4**

*We prove that the probability distribution P' given in Theorem 4 computes the correct marginal probability distribution.*

Here we only give proof for the lower bound. The same approach applies to the upper bound.
¿From the definition of *projection* given above, we know the lower bound for $\pi_U(P)(\bar{x})$ is

$$\min_{I\models P}\Big(\sum_{\bar{y}\in dom(V-U)} I(\bar{x},\bar{y})\Big)$$

.

Here we only give proof for the lower bound. The same approach applies to the upper bound.
¿From the definition of *projection* given above, we know the lower bound for $\pi_U(P)(\bar{x})$ is

$$\min_{I\models P}\Big(\sum_{\bar{y}\in dom(V-U)} I(\bar{x},\bar{y})\Big)$$

.

Let $l'_{(\bar{x})} = \sum_{\bar{y} \in dom(V-U)} l_{(\bar{x},\bar{y})}$, and $u'_{(\bar{x})} = \sum_{\bar{y} \in dom(V-U)} u_{(\bar{x},\bar{y})}$. For any p-interpretation $I$ over $V$ such that $I$ satisifies $P$ ($I \models P$), we have $l_{(\bar{x},\bar{y})} \leq I(\bar{x},\bar{y}) \leq u_{(\bar{x},\bar{y})}$, for all instances $(\bar{x},\bar{y}) \in dom(V)$. Thus the summation over all $\bar{y} \in dom(V-U)$ gives

$$u'_{(\bar{x})} = \sum_{\bar{y} \in dom(V-U)} u_{(\bar{x},\bar{y})} \geq \sum_{\bar{y} \in dom(V-U)} I(\bar{x},\bar{y}) \geq \sum_{\bar{y} \in dom(V-U)} l_{(\bar{x},\bar{y})} = l'_{(\bar{x})}$$

.

By the definition of p-interpretation, we have

$\sum_{(\bar{w},\bar{y}) \in dom(V)} I(\bar{w},\bar{y}) = \sum_{\bar{w} \in dom(U)} \left[ \sum_{\bar{y} \in dom(V-U)} I(\bar{w},\bar{y}) \right] = 1.$

The above equation can be rearranged as follows:

$\sum_{\bar{y} \in dom(V-U)} I(\bar{x},\bar{y}) = 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} \left[ \sum_{\bar{y} \in dom(V-U)} I(\bar{w},\bar{y}) \right].$

By using the relationship given above, $I(\bar{x},\bar{y}) \leq u_{(\bar{x},\bar{y})}$ for any $(\bar{y}) \in dom(V-U)$, we get

$\sum_{\bar{y} \in dom(V-U)} I(\bar{x},\bar{y}) \geq 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} \left[ \sum_{\bar{y} \in dom(V-U)} u_{(\bar{w},\bar{y})} \right] = 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u'_{(\bar{w})}.$

Thus, the lower bound for $\pi_U(P)(\bar{x})$ will be

$$\min_{I \models P} \left( \sum_{\bar{y} \in dom(V-U)} I(\bar{x},\bar{y}) \right) = \max \left( l'_{(\bar{x})}, 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u'_{(\bar{w})} \right)$$

.

On the other hand, the lower bound for $P''(\bar{x})$ is $l''_{(\bar{x})} = \min( \sum_{\bar{y} \in dom(V-U)} l_{(\bar{x},\bar{y})}, 1) = \min(l'_{(\bar{x})}, 1)$, and the upper bound for $P''(\bar{x})$ is $u''_{(\bar{x})} = \min( \sum_{\bar{y} \in dom(V-U)} u_{(\bar{x},\bar{y})}, 1) = \min(u'_{(\bar{x})}, 1)$. By the definition of the tightening operation, we know the lower bound for $\mathcal{T}(P'')(\bar{x})$ is

$$\max \left( l''_{(\bar{x})}, 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u''_{(\bar{w})} \right)$$

.

Now let's prove that for any $\bar{x} \in dom(U)$, the two lower bounds $\max ( l'_{(\bar{x})}, 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u'_{(\bar{w})} )$ and $\max ( l''_{(\bar{x})}, 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u''_{(\bar{w})} )$ are equal for all possible cases. Please note here that $l'_{(\bar{x})} \leq 1$ always holds, and consequently $l''_{(\bar{x})} = l'_{(\bar{x})}$ holds for any $\bar{x} \in dom(U)$.

First we show that if $u'_{(\bar{w})} \leq 1$ holds for all $\bar{w} \in dom(U)$ such that $\bar{w} \neq \bar{x}$, then $u''_{(\bar{w})} = \min(u'_{(\bar{w})}, 1) = u'_{(\bar{w})}$ and therefore $1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u''_{(\bar{w})} = 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u'_{(\bar{x})}$ Then the two lower bound formulas are identical.

To complete the argument that the two lower bounds are equivalent, we also consider the case that there exists $\bar{w} \in dom(U)$, $\bar{w} \neq \bar{x}$ and $u'_{(\bar{w})} > 1$, then $1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u'_{(\bar{x})}$ and $u''_{(\bar{w})} = \min(u'_{(\bar{w})}, 1) = 1$. Therefore $1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u''_{(\bar{w})} \leq 0$. But then,

$\max ( l''_{(\bar{x})}, 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u''_{(\bar{w})} ) = l''_{(\bar{x})} = l'_{(\bar{x})}$, and

$\max ( l'_{(\bar{x})}, 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u'_{(\bar{w})} ) = l'_{(\bar{x})}$.

So the two lower bound expressions will produce the same values.

Therefore, we obtain the following equality:

$\max ( l''_{(\bar{x})}, 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u''_{(\bar{w})} ) = \max ( l'_{(\bar{x})}, 1 - \sum_{\bar{w} \in dom(U) \wedge \bar{w} \neq \bar{x}} u'_{(\bar{w})} )$,

which means that the lower bound for $\mathcal{T}(P'')(\bar{x})$ is equal to the lower bound for $\pi_U(P)(\bar{x})$. □

**Proof of Theorem 5**

*Here we prove that the probability distribution P' given in Theorem 5 computes the correct conditional probability.*

The proof of this theorem is based on the following lemma.

**Lemma 1** *Let $f(x,y,z) = \frac{x}{x+y}$ and the following constraints on $x$, $y$ and $z$ are present:*

*(1)* $x + y + z = 1$
*(2)* $a_1 \le x \le b_1$
*(3)* $a_2 \le y \le b_2$
*(4)* $a_3 \le z \le b_3$

*Let $b_x = \max(x)$ such that $(x,y,z)$ satisfy (1)–(4) and let $a = \min_{x=b_x}(y)$ such that $(x = b_x, y, z)$ satisfy (1)–(4). Then $f(x,y,z)$ reaches its maximum at the point $(b_x, a, 1 - b_x - a)$.*

*Let $a_x = \min(x)$ such that $(x,y,z)$ satisfy (1)-(4) and let $b = \min_{x=a_x}(y)$ such that $(x = a_x, y, z)$ satisfy (1)–(4). Then $f(x,y,z)$ reaches its minimum at the point $(a_x, b, 1 - a_x - b)$.*

**Proof (Lemma 1).**

Without loss of generality, we may assume that the constraints are tight, namely that $b_x = b_1$ and $a_x = a_1$, $y$ reaches both $a_2$ and $b_2$, and $z$ reaches both $a_3$ and $b_3$.

We will prove the lemma for the case of *maximum* of $f(x,y,z)$. The other case is symmetric.

At the point $(b_x = b_1, a, 1 - b_1 - a)$ we have $f(x,y,z) = \frac{b_1}{b_1+a}$. We know that $\min_{x=b_1;(1)-(4)\ \text{satisfied}}(y)$ is the maximum of $a_2$ and $1 - b_1 - b_3$. We consider each of the two cases separately.

1. $a = \min_{x=b_1;(1)-(4)\text{satisfied}}(y) = \max(a_2, 1 - b_1 - b_3) = \mathbf{a_2}$.

   In this case $f(b_1, a_2, 1 - b_1 - a_2) = \frac{b_1}{b_1+a_2}$. We need to show that no other point $(x,y,z)$ that satisfies (1)–(4) produces a larger value for $f$.

   Assume there exists some point $(x', y', z') \neq (b_1, a_2, 1 - b_1 - a_2)$ satisfying (1)–(4), such that $f(x', y', z') = \frac{x'}{x'+y'} > \frac{b_1}{b_1+a_2} = f(b_1, a_2, 1 - b_1 - a_2)$.

   We immediately notice the following:

   - $x' \neq b_1$. If $x' = b_1$ then, $b_1 + a_2 > b_1 + y'$, i.e., $y' < a_2$. However, this is inconsistent with inequality (2).
   - $y' < a_2$. Indeed, if $y' \ge a_2$ while $x' < b_1$ we get the following: $\frac{x'}{x'+y'} \le \frac{x'}{x'+a_2} < \frac{b_1}{b_1+a_2}$.

   However, as $(x', y', z')$ must satisfy (1)–(4) (and in particular, (2)), $y' \ge a_2$, which leads to a contradiction. Therefore, $f$ reaches its maximum at point $(b_1, a_2, 1 - b_1 - a_2)$.

2. $a = \min_{x=b_1;(1)-(4)\text{satisfied}}(y) = \max(a_2, 1 - b_1 - b_3) = \mathbf{1 - b_1 - b_3}$.

   Here $f(b_1, 1 - b_1 - b_3, b_3) = \frac{b_1}{b_1+1-b_1-b_3} = \frac{b_1}{1-b_3}$. We need to show that no other point $(x,y,z)$ that satisfies (1)–(4) produces a larger value for $f$.

   Assume there exists some point $(x', y', z') \neq (b_1, 1 - b_1 - b_3, b_3)$ satisfying (1)–(4), such that $f(x', y', z') = \frac{x'}{x'+y'} > \frac{b_1}{1-b_3} = f(b_1, 1 - b_1 - b_3, b_3)$.

   By reasoning similar to the previous case we can show that $x' < b_1$ here as well. Then, since $\frac{x'}{x'+y'} > \frac{b_1}{1-b_3}$ and both denominators are strictly positive, $(1 - b_3) \cdot x' > (x' + y') \cdot b_1$ so $\frac{(1-b_1-b_3)}{b_1} \cdot x' > y'$.

   We will show that no point $(x', y', z')$ can satisfy (1)–(4), $x' < b_1$ and $\frac{(1-b_1-b_3)}{b_1} \cdot x' > y'$ at the same time. Consider $y' = \frac{(1-b_1-b_3)}{b_1} \cdot x'$. Then, substituting into (1), we get: $x' + \frac{(1-b_1-b_3)}{b_1} \cdot x' + z' = 1$, which leads, after simplifications, to $z' = 1 - \frac{1-b_3}{b_1} \cdot x'$. By (4), $a_3 \le z' \le b_3$. Substituting $z'$ in the right side of the inequality we get $1 - \frac{1-b_3}{b_1} \cdot x' \le b_3$ i.e., $1 - b_3 \le \frac{1-b_3}{b_1} \cdot x'$. But $x' < b_1$, $\frac{x'}{b_1} < 1$ and hence $\frac{1-b_3}{b_1} \cdot x' < (1 - b_3) \cdot 1 = 1 - b_3$, which leads to a contradiction.

   Therefore, $f$ reaches its maximum among all points satisfying (1)–(4), at point $(b_1, 1 - b_1 - b_3, b_3)$. □

Now let's prove **Theorem 5**.
We prove the theorem for lower bound $l'_{\bar{y}}$ of $P'(\bar{y})$.

As follows from Definition 29 of conditionalization,

$$l_{\bar{y}} = \min_{I \models P} \left( \frac{I_X(\bar{y})}{\sum_{\bar{y}' \in dom(V')} I_X(\bar{y}')} \right).$$

Notice that $\sum_{\bar{y}' \in dom(V')} I_X(\bar{y}')$ can be rewritten as $I_X(\bar{y}) + \sum_{\bar{y}' \neq \bar{y}} I_X(\bar{y}')$, and therefore,

$$l_{\bar{y}} = \min_{I \models P} \left( \frac{I_X(\bar{y})}{I_X(\bar{y}) + \sum_{\bar{y}' \neq \bar{y}} I_X(\bar{y}')} \right).$$

Given $\bar{y} \in dom(V')$ and $X \subset dom(v)$ we can separate all vectors in $dom(V)$ into three disjoint sets:
$W_{\bar{y}} = \{(\bar{y}, x) | x \in X\}$,
$Y_{\bar{y}} = \{(\bar{y}', x) | \bar{y}' \neq \bar{y} \wedge x \in X\}$, and
$Z_{\bar{y}} = \{(\bar{y}', x') | x' \notin X\}$.

Given a p-interpretation $I$, we notice that

$\sum_{\bar{w} \in W_{\bar{y}}} I(\bar{w}) = I_X(\bar{y})$;
$\sum_{\bar{w} \in Y_{\bar{y}}} I(\bar{w}) = \sum_{\bar{y}' \in dom(V'), \bar{y}' \neq \bar{y}} I_X(\bar{y}')$;
(1) $\sum_{\bar{w} \in W_{\bar{y}}} I(\bar{w}) + \sum_{\bar{w} \in Y_{\bar{y}}} I(\bar{w}) + \sum_{\bar{w} \in Z_{\bar{y}}} I(\bar{w}) = \sum_{\bar{w} \in dom(V)} I(\bar{w}) = 1$.

If $I \models P$ then we also get

(2) $\sum_{\bar{w} \in W_{\bar{y}}} l_{\bar{w}} \leq \sum_{\bar{w} \in W_{\bar{y}}} I(\bar{w}) \leq \sum_{\bar{w} \in W_{\bar{y}}} u_{\bar{w}}$,
(3) $\sum_{\bar{w} \in Y_{\bar{y}}} l_{\bar{w}} \leq \sum_{\bar{w} \in Y_{\bar{y}}} I(\bar{w}) \leq \sum_{\bar{w} \in Y_{\bar{y}}} u_{\bar{w}}$, and
(4) $\sum_{\bar{w} \in Z_{\bar{y}}} l_{\bar{w}} \leq \sum_{\bar{w} \in Z_{\bar{y}}} I(\bar{w}) \leq \sum_{\bar{w} \in Z_{\bar{y}}} u_{\bar{w}}$.

Replacing $\sum_{\bar{w} \in W_{\bar{y}}} I(\bar{w}) = I_X(\bar{y})$ with $x$, $\sum_{\bar{w} \in Y_{\bar{y}}} I(\bar{w}) = \sum_{\bar{y}' \neq \bar{y}} I_X(\bar{y}')$ with $y$ and $\sum_{\bar{w} \in W_{\bar{y}}} I(\bar{w})$ with $z$, the problem of determining $l_{\bar{y}} = \min_{I \models P} \left( \frac{I(\bar{y}, x)}{I(\bar{y}, x) + \sum_{\bar{y}' \in dom(V'), \bar{y}' \neq \bar{y}} I(\bar{y}', x)} \right)$ reduces to the problem of minimizing $f(x, y, z) = \frac{x}{x+y}$ subject to constraints (1)–(4) (or, to be more exact, their transformed versions).

By Lemma 1 we know that the minimum of $\frac{x}{x+y}$ on the set specified by constraints (1)–(4) will occur at the point that minimizes the value of $x$ ($I_X(\bar{y})$) (subject to (1)–(4)), and then maximizes $y$ given that $x$ is at its minimum.

By Theorem 2, the minimal value of $I_X(\bar{y})$ is $\max(\sum_{x \in X} l_{(\bar{y}, x)}, 1 - \sum_{\bar{y}' \neq \bar{y} \vee x' \notin X} u_{(\bar{y}', x')}) = l[X]_{\bar{y}}$.

Similarly, given that $I_X(\bar{y}) = l[X]_{\bar{y}}$, we have that

$$\max(\sum_{\bar{y}' \neq \bar{y}} I_X(\bar{y}')) = \min(1 - \sum_{x' \notin X} l_{(\bar{y}', x')}, \sum_{\bar{y}' \neq \bar{y}, x \in X} u_{(\bar{y}', x)} + l[X]_{\bar{y}}).$$

Therefore,

$$l'_{\bar{y}} = \frac{l[X]_{\bar{y}}}{\min\left(1 - \sum_{x' \notin X} l_{(\bar{y}', x')}\right)}.$$

□