

Possibilistic Conditional Tables

Olivier Pivert¹ and Henri Prade^{2,3}(✉)

¹ Irista – Enssat, University of Rennes 1, Technopole Anticipa,

22305 Lannion Cedex, France

pivert@enssat.fr

² IRIT, CNRS and University of Toulouse, 31062 Toulouse Cedex 9, France

³ QCIS, University of Technology, Sydney, Australia

prade@irit.fr

Abstract. On the one hand possibility theory and possibilistic logic offer a powerful representation setting in artificial intelligence for handling uncertainty in a qualitative manner. On the other hand conditional tables (*c*-tables for short) and their probabilistic extension provide a well-known setting for representing respectively incomplete and uncertain information in relational databases. Although these two settings rely on the idea of possible worlds, they have been developed and used independently. This paper investigates the links between possibility theory, possibilistic logic and *c*-tables, before introducing possibilistic *c*-tables and discussing their relation with a recent certainty-based approach to uncertain databases and their differences with probabilistic *c*-tables.

1 Introduction

The representation and the handling of imperfect information, be it incomplete or uncertain, has led for several decades to the development of important research trends in artificial intelligence, such as nonmonotonic reasoning, belief revision, reasoning under uncertainty, or information fusion. Different formalisms have been introduced for representing uncertainty which depart from probability theory, and which are particularly of interest when information is both incomplete and uncertain, such as belief function theory or possibility theory. Moreover, possibility theory [11] may have a qualitative flavor, which may be appropriate when uncertainty is hard to assess precisely.

On its side, database research first tackled the issue of managing imperfect/incomplete information a long time ago, with the works of Codd [7] and Lipski [20] in the late seventies. Researchers first focused their attention on so-called *null* values (either totally unknown or not applicable), before considering more sophisticated cases of incompleteness. The pioneering work by Imielinski and Lipski [18] considers three models: Codd tables, *v*-tables and *c*-tables. *v*-tables are conventional instances where variables can appear in addition to constants from the considered domains. Codd tables are *v*-tables in which all the variables are distinct. They correspond roughly to the current use of nulls in SQL, while *v*-tables model “labeled” or “marked” nulls. As to *c*-tables, they are

v -tables extended with conditions on variables and constitute a very powerful model, even though they raise computational complexity issues.

The last decade witnessed a renewal of interest in the modeling and management of uncertain databases (in particular, probabilistic ones, see, e.g., [27]), which brought the c -tables model back in the spotlights. As pointed out by Antova et al. in [3], the three most prominent probabilistic database models proposed recently, namely ULDBs [4, 5], U-relations [3] and the model proposed in [8] based on virtual views, are different versions of the concept of a c -table. See also the works by Green and Tannen who proposed a probabilistic c -tables model [17] and by Shimizu et al. [26].

Even though most of the literature about uncertain databases uses probability theory as the underlying uncertainty model, this type of modeling is not always so easy, as recognized in the introductory chapter of [27]: “Where do the probabilities in a probabilistic database come from? And what exactly do they mean? The answer to these questions may differ from application to application, but it is rarely satisfactory.” This is one of the reasons why some authors have proposed approaches that rather rest on another model, namely possibility theory [10, 28], which is appropriate for representing epistemic uncertainty.

The idea of applying possibility theory to the modeling of uncertain databases goes back to the early 1980’s [23–25]. At that time, the approach was to represent ill-known attribute values by possibility distributions, and then given a query, to compute the fuzzy set of answers that are *certain* (or sure) to some extent, and the larger fuzzy set of answers that are *possible* to some extent. This was an attempt at providing a *graded* counterpart to the modal logic-based approach proposed by Lipski [20, 21]. This possibility distribution-based representation of the information available about attribute values was covering the cases of null values, of classical subset restrictions on possible values, and more generally of fuzzy subset restrictions when all possible values are not equally plausible; it had nothing to do with the notion of c -tables. Note also that possibilistic logic [9], which is useful for having a possibilistic reading of c -tables as shown in the following, has been introduced later. Then the possibility distribution-based representation of ill-known attribute values has become the standard approach in possibility theory for handling databases with missing, imprecise, or uncertain information until now.

Recent advances on this approach can be found in [6]. In contrast with probability theory, one expects the following advantages when using possibility theory:

- the qualitative nature of the model makes easier the elicitation of the degrees or levels attached to candidate values, inasmuch as an ordinal scale \mathcal{L} made of $k + 1$ linguistic labels may be used to assess the certainty (and possibility) levels attached to an attribute value or a tuple. For instance, with $k = 4$, one may use:

$$\begin{aligned} \alpha_0 &= \text{“not at all”} < \alpha_1 = \text{“somewhat”} < \\ \alpha_2 &= \text{“rather”} < \alpha_3 = \text{“almost”} < \alpha_4 = \text{“totally”} \end{aligned}$$

where α_0 (resp. α_k) corresponds to 0 (resp. 1) in the unit interval when a numerical framework is used. In possibility theory [10] the necessity

(or certainty) $N(E)$ of an event E is defined as the impossibility of the opposite event, i.e., $N(E) = 1 - \Pi(\bar{E})$. Then the operation $1 - (\cdot)$ that is used when the degrees belong to the unit interval is replaced by the order reversal operation denoted by $rev(\cdot)$: $rev(\alpha_i) = \alpha_{k-i}$. In the following, however, we use the numerical scale $[0, 1]$ in order not to make the formulas too cumbersome.

- in probability theory, the fact that the sum of the degrees from a distribution must equal 1 makes it difficult to deal with incompletely known distributions.

More recently, the authors of the present paper developed a new idea which is to use the notion of necessity from possibility theory to qualify the *certainty* that an ill-known piece of data takes a given value [22]. In contrast with both probabilistic databases *and* possibilistic ones in the sense of [6, 25], — which can be referred to as the “full possibilistic” approach —, the main advantage of the certainty-based model lies in the fact that operations from relational algebra can be extended in a simple way and with a data complexity which is the same as in a classical database context (i.e., when all data are certain).

However, the model defined in [22] does not yield the same expressive power as c -tables inasmuch as it only aims to represent the more or less certain values (or disjunction of values) that some attributes in a tuple may take, but does not consider conditions attached to tuples as c -tables do. Here, our main objective is twofold (i) first to establish the link between c -tables and possibilistic representations, two settings that have remained unrelated in spite of their close concerns, and (ii) to propose a possibilistic counterpart of the probabilistic c -tables model defined in [17], which will make it possible to deal with more complex cases of imprecise and uncertain information. The objective of this first paper relating c -tables and possibility theory is to discuss the ideas underlying the two settings, illustrating them on suitable examples, and identifying the key principles for bridging them. More precisely, we intend to:

- show how regular c -tables can be interpreted, then extended, in the setting of possibilistic logic;
- establish the link between the extended relational model presented in [22], based on possibilistic certainty, and that of possibilistic c -tables.

The remainder of the paper is structured as follows. Section 2 is devoted to a presentation of the c -tables model. Section 3 provides a refresher on possibility theory. Section 4 presents the concept of a possibilistic c -table. It first shows how regular c -tables may be interpreted in terms of possibilistic logic (in the particular case where necessity degrees equal 0 or 1), then describes a gradual extension, discusses the certainty-based model as a particular case, and finally compare possibilistic c -tables and probabilistic ones as introduced in [17]. Finally, Sect. 5 recalls the main contributions and outlines some research perspectives.

2 Refresher on c -tables

As mentioned above, conditional tables (c -tables for short) are v -tables extended with conditions on variables. Let us recall the definition from [18].

Table 1. Conditional table from Example 1

<i>Student</i>	<i>Course</i>	
		$(x \neq \text{math}) \wedge (x \neq \text{CS})$
Sally	math	$(z = 0)$
Sally	CS	$(z \neq 0)$
Sally	x	
Alice	biology	$(z = 0)$
Alice	math	$(x = \text{physics}) \wedge (t = 0)$
Alice	physics	$(x = \text{physics}) \wedge (t \neq 0)$

Let \mathcal{U} be a fixed, finite set of *attributes*. Attributes are denoted by A, B, C , and sets of attributes by X, Y, Z . Associated with every $A \in \mathcal{U}$ is an *attribute domain* $D(A)$. We denote $D = \bigcup_{A \in \mathcal{U}} D(A)$. Elements of D are called *constants*. For every $A \in \mathcal{U}$, let $V(A)$ be a countably infinite set of symbols called *variables*. It is assumed that $V(A) \cap D = \emptyset$, $V(A) \cap V(B) = \emptyset$ if $D(A) \neq D(B)$ and $V(A) = V(B)$ if $D(A) = D(B)$. Let us denote by \mathcal{S} the set of all expressions built up from atomic conditions of the form $(x = a)$, $(x = y)$, *false* and *true*, where for some $A \in \mathcal{U}$, $a \in D(A)$, $x, y \in V(A)$, by means of \neg , \vee , and \wedge . In the following, we use the notation $x \neq a$ (resp. $x \neq y$) for $\neg(x = a)$ (resp. $\neg(x = y)$). For every condition $\Phi \in \mathcal{S}$, we say that a valuation v satisfies Φ if its assignment of constants to variables makes the formula true.

Definition 1. By a *c-tuple* on a set of attributes X , we mean any mapping t defined on $X \cup \{\text{con}\}$ such that $t(X)$ is a V -tuple (i.e., any mapping t' that associates an element $t'(A) \in D(A) \cup V(A)$ with every $A \in X$) and $t(\text{con}) \in \mathcal{S}$ is the condition associated with $t(X)$. A *conditional table* (or *c-table*) on X is any finite set T of *c-tuples* on X .

In this original definition, conditions may only be attached to individual tuples. An extension is to consider that a condition may also be attached to the *c-table* globally, see [1]. The definition then becomes:

Definition 2. A *conditional table* (or *c-table*) on X is a pair (T, Φ_T) where T is any finite set of *c-tuples* on X and $\Phi_T \in \mathcal{S}$ is a global condition on T .

In the following, we denote by φ_t the condition associated with tuple t .

Example 1. Suppose we know that Sally is taking math or computer science (CS) (but not both) and another course; Alice takes biology if Sally takes math, and math or physics (but not both) if Sally takes physics. This can be represented by the *c-table* depicted in Table 1.

Remark 1. Notice that it is possible to use disjunctions that are not mutually exclusive. For instance, if Sally could take math or physics or both, the first two lines of Table 1 would have to be replaced by the four lines:

Sally math $x = 0$
Sally CS $x = 0$
Sally math $x \neq 0 \wedge y = 0$
Sally CS $x \neq 0 \wedge y \neq 0$

□

Adopting the closed world assumption, a given c -table T represents a set of instances (possible worlds) as follows:

$$\text{rep}(T) = \{I \mid \text{there is a valuation } \nu \text{ satisfying } \Phi_T \text{ such that relation } I \text{ consists exactly of those facts } \nu(t) \text{ for which } \nu \text{ satisfies } \varphi_t\}.$$

In the context of uncertain databases, the notion of a *strong representation system* plays an important part. Let us recall its definition [1]. Consider some particular representation system (e.g., tables). Such a system involves a language for describing representations and a mapping rep that associates a set of instances with each representation. Suppose that we are interested in a particular query language \mathcal{L} (e.g., relational algebra). We would like to be capable of representing the result of a query in the same system. More precisely, for each representation T and query q , there should exist a computable representation $\bar{q}(T)$ such that

$$\text{rep}(\bar{q}(T)) = q(\text{rep}(T)). \quad (1)$$

In other words, $\bar{q}(T)$ represents the possible results of q , i.e., $\{q(I) \mid I \in \text{rep}(T)\}$. In such a context, a *possible answer* (resp. a *certain answer*) to a query q is a tuple that belongs to at least one world (resp. to all of the worlds) of $\bar{q}(T)$:

$$t \in \text{poss}(q) \Leftrightarrow t \in \bigcup_{I \in \text{rep}(\bar{q}(T))} I \quad (2)$$

$$t \in \text{cert}(q) \Leftrightarrow t \in \bigcap_{I \in \text{rep}(\bar{q}(T))} I \quad (3)$$

Example 2. If we consider Table 1, the possible (resp. certain) answers to the query “find the students who take math or computer science” are {Sally, Alice} (resp. {Sally}). ◊

If some representation system τ possesses property (1) for a query language \mathcal{L} , then τ is said to be a *strong representation system* for \mathcal{L} . It has been proven that c -tables are a strong representation system for relational algebra [18]. For each operation u of relational algebra, [18] defines an operation \bar{u} on c -tables. Hereafter, we recall these definitions using the easier-to-read presentation from [17]. For projection, we have

$$\bar{\pi}_\ell(T) := \{(t' : \varphi_{t'}) \mid t \in T \text{ s.t. } \pi_\ell(t) = t', \varphi_{t'} = \bigvee \varphi_t\}$$

Table 2. Conditional table T (left) and result of $\sigma_{B \neq b}(T)$ (right)

B	C
y	$c \quad (y \neq b)$
z	w
b	x

B	C
y	$c \quad (y \neq b)$
z	$w \quad (z \neq b)$

Table 3. Conditional table from Example 3

<i>Course</i>	<i>Room</i>
<i>math</i>	<i>Fermat</i> ($u = 0$)
<i>math</i>	<i>Gödel</i> ($u \neq 0$)
<i>CS</i>	<i>Turing</i> ($v \neq 0$)
<i>CS</i>	<i>Gödel</i> ($v = 0$) \wedge ($u = 0$)
<i>CS</i>	<i>Fermat</i> ($v = 0$) \wedge ($u \neq 0$)
<i>biology</i>	<i>Turing</i> ($v = 0$)
<i>biology</i>	<i>Monod</i> ($v \neq 0$)
<i>physics</i>	<i>Einstein</i>

where ℓ is a list of indexes and the disjunction is over all t in T such that $\pi_\ell(t) = t'$. For selection, we have

$$\bar{\sigma}_c(T) := \{(t : \varphi_t) \wedge c(t) \mid (t, \varphi_t) \in T\}$$

where $c(t)$ denotes the result of evaluating the selection predicate c on the values in t (this is in general a Boolean formula on constants and variables). See Table 2 for an example.

For cross product and union, we have

$$T_1 \bar{\times} T_2 := \{(t_1 \times t_2 : \varphi_{t_1} \wedge \varphi_{t_2}) \mid t_1 \in T_1, t_2 \in T_2\}$$

$$T_1 \bar{\cup} T_2 := T_1 \cup T_2$$

Difference and intersection are handled similarly.

Example 3. Here is an example involving a join between two conditional tables. Let us assume that math courses take place either in Fermat room or in Gauss room. The CS course takes place in Turing room if the biology course does not use it, otherwise it takes place in Gödel room or in Fermat room. The course either takes place in Monod room or in Turing room. Physics is in Einstein room.

We are interested in knowing in which room each student may attend a course, which implies joining Tables 1 and 3. The result is represented in Table 4. For instance, the first line expresses that Sally will be in Fermat room if she takes math and math is in Fermat. \diamond

It is quite clear that c -tables *per se* are quite difficult to interpret by end-users, but they can be used as a means to answer yes/no queries. Complexity results about different types of yes/no queries on c -tables are given in [2, 15]. The authors consider five types of problems:

Table 4. Resulting conditional table

<i>Course</i>	<i>Room</i>	
<i>Sally</i>	<i>Fermat</i>	$(z = 0) \wedge (u = 0)$
<i>Sally</i>	<i>Gödel</i>	$(z = 0) \wedge (u \neq 0)$
<i>Sally</i>	<i>Turing</i>	$(z \neq 0) \wedge (v \neq 0)$
<i>Sally</i>	<i>Monod</i>	$x = \textit{biology} \wedge (v \neq 0)$
<i>Sally</i>	<i>Einstein</i>	$x = \textit{physics}$
<i>Alice</i>	<i>Fermat</i>	$(x = \textit{physics}) \wedge (t \neq 0) \wedge (u = 0)$
<i>Alice</i>	<i>Gödel</i>	$(x = \textit{physics}) \wedge (t \neq 0) \wedge (u \neq 0)$
<i>Alice</i>	<i>Turing</i>	$(z = 0) \wedge (v = 0)$
<i>Alice</i>	<i>Monod</i>	$(z = 0) \wedge (v \neq 0)$
<i>Alice</i>	<i>Einstein</i>	$(x = \textit{physics}) \wedge (t \neq 0)$

- containment problem $\textit{cont}(q_0, q)$: is the result of a given query q_0 over a given set of c -tables \mathcal{S}_0 included in the result of a given query q over a given set of c -tables \mathcal{S} ?
- membership problem $\textit{memb}(q)$: is a given instance I_0 a possible world of the result of a given query q over a given set of c -tables \mathcal{S} ?
- uniqueness problem $\textit{uniq}(q_0)$: is the result of a given query q_0 equal to the single given possible world $\{I\}$?
- possibility problem $\textit{poss}(*, q)$: do all the facts of a given set P belong to a same possible world of the result of a given query q over a given set of c -tables \mathcal{S} ?
- certainty problem $\textit{cert}(*, q)$: do all the facts of a given set P belong to every possible world of the result of a given query q over a given set of c -tables \mathcal{S} ?

In [2, 15], the authors show that for any polynomial time computable queries q_0, q , one has:

- $\textit{cont}(q_0, q)$ is in Π_2^P ;
- $\textit{memb}(q)$ is in NP;
- $\textit{uniq}(q_0)$ is in coNP;
- $\textit{poss}(*, q)$ is in NP; and
- $\textit{cert}(*, q)$ is in coNP.

3 Refresher on Possibility Theory

In possibility theory [10, 28], each event E — defined as a subset of a universe Ω — is associated with two measures, its possibility $\Pi(E)$ and its necessity $N(E)$. Π and N are two dual measures, in the sense that

$$N(E) = 1 - \Pi(\overline{E})$$

(where the overbar denotes complementation). This clearly departs from the probabilistic situation where $\textit{Prob}(E) = 1 - \textit{Prob}(\overline{E})$. So in the probabilistic case, as soon as you are not certain about E ($\textit{Prob}(E)$ is small), you become rather

certain about \bar{E} ($Prob(\bar{E})$ is large). This is not at all the situation in possibility theory, where complete ignorance about E ($E \neq \emptyset, E \neq \Omega$) is allowed: This is represented by $\Pi(E) = \Pi(\bar{E}) = 1$, and thus $N(E) = N(\bar{E}) = 0$. In possibility theory, being somewhat certain about E ($N(E)$ has a high value) forces you to have \bar{E} rather impossible ($1 - \Pi$ is impossibility), but it is allowed to have no certainty neither about E nor about \bar{E} . Generally speaking, possibility theory is oriented towards the representation of epistemic states of information, while probabilities are deeply linked to the ideas of randomness, and of betting in case of subjective probability, which both lead to an additive model such that $Prob(E) = 1 - Prob(\bar{E})$.

A possibility measure Π (as well as its dual necessity measure N) is based on a possibility distribution π , which is a mapping from a referential U to an ordered scale, say $[0, 1]$. Namely,

$$\Pi(E) = \sup_{u \in E} \pi(u) \text{ and } N(E) = \inf_{u \notin E} (1 - \pi(u)).$$

$\pi(u) = 0$ means that u is (fully) impossible, while $\pi(u) = 1$ just means u is (fully) possible, since it is important to notice that nothing prevents to have $u \neq u'$ and $\pi(u) = \pi(u') = 1$. Thus, $E = \{u\}$ is (fully) certain only if $\pi(u) = 1$ and $\forall u' \neq u, \pi(u') = 0$. A possibility distribution π is normalized as soon as $\exists u, \pi(u) = 1$; it expresses a form of consistency, since it is natural to have at least one alternative fully possible as soon as the referential is exhaustive (this is the counterpart in possibility theory of having the sum of the probabilities in a probability distribution equal to 1).

Conversely, if we know that $N(E) \geq \alpha$, which means that we are certain (at least) at level α that E is true, there are several possibility distributions that can be compatible with this constraint, but it can be shown that the largest one (the one that allocates the greatest possible possibility to each $u \in U$) is unique, and is such that $\pi(u) = 1$ if $u \in E$ and $\pi(u) = 1 - \alpha$ if $u \notin E$. So, if we are α -certain that Bob lives in Paris or Lyon, this is represented by the distribution $\pi(Paris) = 1 = \pi(Lyon)$, and $\pi(u) = 1 - \alpha$ for any other city u .

Representing a possibility distribution with more than two levels in terms of constraints of the form $N(E) \geq \alpha$ requires several constraints. For instance, if it is possible that Peter lives in Brest, Paris, Lyon, or another city with respective possibility levels $1 > \alpha > \alpha' > \alpha''$ (i.e., $\pi(Brest) = 1, \pi(Paris) = \alpha, \pi(Lyon) = \alpha', \pi(u) = \alpha''$ for any other city u), then, it corresponds to the constraints $N(\{Brest, Paris, Lyon\}) \geq 1 - \alpha'', N(\{Brest, Paris\}) \geq 1 - \alpha'$ and $N(\{Brest\}) \geq 1 - \alpha$. More generally, any possibility distribution with a finite number of levels $1 = \alpha_1 > \dots > \alpha_n > 0 = \alpha_{n+1}$ can be represented by a collection of n constraints of the form $N(E_i) > 1 - \alpha_{i+1}$ with $E_i = \{u \mid \pi(u) \geq \alpha_i\}$.

Constraints such as $N(E) \geq \alpha$ where E stands for the set of models of a proposition p can be handled inside possibilistic logic under the form of a pair (p, α) made of the classical logic proposition p and a level α belonging to a linearly ordered scale [9, 12]. The semantics of a possibilistic logic base K , i.e.,

a set of such a pairs, $K = \{(p_i \alpha_i) | i = 1, \dots, n\}$ is expressed by means of the possibility distribution π_K defined by

$$\forall \omega \in \Omega, \pi_K(\omega) = \min_{i=1, \dots, n} \pi_{\{(p_i \alpha_i)\}}(\omega) \text{ with } \pi_{\{(p_i \alpha_i)\}}(\omega) = \max([p_i](\omega), 1 - \alpha_i)$$

where Ω is the set of interpretations of the language induced by the literals of the formulas in K , and $[p_i](\omega) = 1$ if $\omega \models p_i$ (i.e., ω is a model of K) and $[p_i](\omega) = 0$ otherwise. The semantic entailment is then defined by

$$K \models (p, \alpha) \text{ if and only if } N_K([p]) \geq \alpha \Leftrightarrow \forall \omega \pi_K(\omega) \leq \max([p](\omega), 1 - \alpha)$$

where N_K is the necessity measure defined from π_K .

The syntactic inference machinery of possibilistic logic, using the resolution rule

$$(\neg p \vee q, \alpha), (p \vee \neg r, \beta) \vdash (q \vee \neg r, \min(\alpha, \beta))$$

and refutation (it amounts to adding $(\neg\varphi, 1)$, put in clausal form, to K , and using this rule repeatedly to show that $K \cup (\neg\varphi, 1) \vdash (\perp, a)$), has been proved to be sound and complete with respect to the semantics. Algorithms and complexity evaluation (similar to the one of classical logic) can be found in [19]. It is worth mentioning that the repeated use of the probabilistic resolution rule $P(p \vee q) \geq \alpha, P(\neg p \vee r) \geq \beta \vdash P(q \vee r) \geq \max(0, \alpha + \beta - 1)$ does not always provide the tightest bounds that can be obtained by probability computation and thus does not lead to a complete calculus. Moreover, a mixed resolution rule that involves necessity and possibility bounds holds in possibilistic logic [9], here written in terms of semantic constraints:

$$N(\neg p \vee q) \geq \alpha, \Pi(p \vee \neg r) \geq \beta \models \Pi(q \vee \neg r) \geq \beta \text{ provided that } \alpha > 1 - \beta.$$

Lastly, it is worth mentioning that a formula such as $(\neg p \vee q, \alpha)$ is semantically equivalent to $(q, \min(\alpha, [p]))$ (with $[p] = 1$ if p is true and $[p] = 0$ if p is false), where $(q, \min(\alpha, [p]))$ can be read q is α -certain provided that p is true. Then it can be checked that the following resolution rule holds $(q, \min(\alpha, [p])), (p, \min(\beta, [r])) \vdash (q, \min(\alpha, \beta, [r]))$. The interested reader may find more details about possibilistic logic and its various applications in [12].

4 Possibilistic c-tables

In his pioneering work [20, 21] on databases with incomplete information Lipski distinguishes between answers that are certain and answers that are only possible for a given query in a modal logic setting. Inspired by this work, the use of possibility theory for modeling incomplete and uncertain information in databases was first proposed in [23–25], and later revisited in [6]. In these approaches, each attribute value in a tuple is represented by means of a possibility distribution *defined on the domain of the attribute*. This possibility distribution restricts the more or less possible values of the attribute for the considered tuple according to the available information (as, e.g., in the previous section example of Peter

Table 5. Conditional table from Example 4

<i>Student</i>	<i>Course</i>	
<i>Sally</i>	<i>math</i>	$(z = 0)$
<i>Sally</i>	<i>CS</i>	$(z \neq 0)$

living in Brest, Paris, Lyon, or another city). However these works were making no reference to c -tables. In the following, we show how any c -table can be directly understood as a possibility distribution over a set of possible database worlds, and thus expressed by a possibilistic logic database. We first do it in the case where possibility is not graded, and then consider the general case of graded possibility that enables us to accommodate uncertainty. We then relate possibilistic c -tables to the particular case of the certainty-based approach to uncertain databases that we recently developed, and finally compare possibilistic c -tables to probabilistic c -tables.

4.1 Regular c -tables Interpreted in Terms of Possibilistic Logic

As already mentioned, the conditions associated with tuples in a c -table are specifying possible worlds. Thus, they implicitly define possibility distributions over mutually exclusive situations.

Example 4. If we take the first part of Example 1, namely representing the information $(take(Sally, math) \vee take(Sally, CS))$, this corresponds to the possibility distribution $\pi_{take(Sally, \cdot)}(math) = 1 = \pi_{take(Sally, \cdot)}(CS)$, or if we prefer $\pi(z = 0) = \pi(z \neq 0) = 1$. Note that the situation in Remark 1 where $math$ and CS are no longer mutually exclusive would require a possibility distribution defined on the power set of $\{math, CS\}$. Still both cases can be easily captured in possibilistic logic. Indeed “Sally is taking math or computer science” is expressed by

$$(take(Sally, math) \vee take(Sally, CS), 1)$$

and the additional constraint “but not both” by

$$(\neg take(Sally, math) \vee \neg take(Sally, CS), 1).$$

Let us now examine the rest of Example 1. We can take for the domain of attribute *Course* the set $D_{Course} = \{math, CS, biology, physics, others\}$ that involves all the topics mentioned in the example and leave room for others. Then the information “Sally takes another course” (apart from “math” or “CS”) writes in possibilistic logic

$$(take(Sally, physics) \vee take(Sally, biology) \vee take(Sally, others), 1)$$

while “Alice takes biology if Sally takes math and math or physics (but not both) if Sally takes physics” writes

$$\begin{aligned} &(take(Alice, biology), [take(Sally, math)]) \\ &(take(Alice, math) \vee take(Alice, physics), [take(Sally, physics)]) \\ &(\neg take(Alice, math) \vee \neg take(Alice, physics), 1). \end{aligned}$$

We could equivalently write $(take(Alice, biology) \vee \neg take(Sally, math), 1)$ in place of $(take(Alice, biology), [take(Sally, math)])$, and this applies as well to possibilistic formula after. Having $[take(Sally, math)]$ (or $[take(Sally, physics)]$) in the certainty level slot of the formulas puts the main focus on Alice. \diamond

Thus, the conditional table represented in Table 5 translates easily in a possibilistic logic base. This base is semantically associated to a possibility distribution, which can be obtained by applying the general result recalled in Sect. 3. This would enable us to explicit the possibility distribution underlying Table 5. This is a $\{0, 1\}$ -valued possibility distribution; however, if binary-valued functions are used in the certainty slots (as $[take(Sally, math)]$ in the above example), some possibility degrees will receive a conditional value (such as $1 - [take(Sally, math)] \in \{0, 1\}$).

A query such as “find the x ’s such that condition Q is true” is processed by refutation, adding the formulas corresponding to $\neg Q(x) \vee answer(x)$ to the base, using a small trick due to [16] (see [9]). Let us take a first example.

Example 5. In the previous Example 4, let us consider the query “find the students who take math or computer science”, which translates into

$$\{(\neg take(x, math) \vee answer(x), 1), (\neg take(x, CS) \vee answer(x), 1)\}.$$

It can be checked that it yields $(answer(Sally), 1)$

and $(answer(Alice) \vee take(Alice, physics), [take(Sally, physics)])$ (or equivalently $(answer(Alice), \min([\neg take(Alice, physics)], [take(Sally, physics)]))$)
 $\Leftrightarrow (answer(Alice), [\neg take(Alice, physics)] \wedge [take(Sally, physics)])$. \diamond

As can be seen, we may obtain two types of answers: (i) the answers x_0 of the form $(answer(x_0), 1)$, and (ii) the answers x_0 of the form $(answer(x_0), [\varphi])$ where $[\varphi]$ is a nontautological condition which may take values 0 and 1. The first answers are exactly those that are *certain*, while the second ones are exactly those that are *possible* without being *certain*. Let us consider an example with a join.

Example 6. Let us come back to Example 3 involving a join query. The translation in possibilistic logic is straightforward:

$$\begin{aligned} &(place(physics, Einstein), 1) \\ &(place(math, Fermat) \vee place(math, Gödel), 1), \\ &(\neg place(math, Fermat) \vee \neg place(math, Gödel), 1) \\ &(place(biology, Turing) \vee place(biology, Monod), 1), \\ &(\neg place(biology, Turing) \vee \neg place(biology, Monod), 1) \\ &(\neg place(CS, Turing) \vee \neg place(biology, Turing), 1) \\ &(\neg place(biology, Turing) \vee \neg place(math, Fermat) \vee (place(CS, Gödel), 1) \\ &(\neg place(biology, Turing) \vee \neg place(math, Gödel) \vee (place(CS, Fermat), 1) \\ &(place(CS, Turing) \vee place(CS, Gödel) \vee (place(CS, Fermat), 1) \end{aligned}$$

Let us now consider the question “who is in Monod room?”, which translates into $(\neg take(x, y) \vee \neg place(y, Monod) \vee answer(x), 1)$. Then from the possibilistic logic counterparts of Tables 1 and 3, it can be checked that we can infer

Table 6. Possibilistic conditional tables from Example 7 (r left, s right)

<i>Student</i>	<i>Course</i>	
$x = 0$ (1) \oplus $x = 1$ (0.6)		
<i>Sally</i>	<i>CS</i>	$x = 0$
<i>Sally</i>	<i>math</i>	$x = 1$

<i>Course</i>	<i>Room</i>	
$y = Fermat$ (1) \oplus $y = Turing$ (0.3)		
<i>math</i>	<i>y</i>	
<i>CS</i>	<i>Turing</i>	

$\{Sally, Alice\}$, as the set of possible answers. Indeed we get

$$(answer(Sally), \min([\neg take(Sally, physics)], [\neg take(Sally, others)], [\neg place(biology, Turing)]))$$

$(answer(Alice), \min([take(Sally, math)], [\neg place(biology, Turing)]))$
 which is in agreement with Table 4.

As expected, the conjunctive structure of the combined certainty levels reflects the conjunctions performed when the join of the two c -tables are computed. \diamond

4.2 Gradual Possibilistic c -tables

Obviously, we are not obliged to use binary-valued possibility and certainty levels taking values ‘0’ or ‘1’ only. We can thus express for instance that “Sally is taking math or computer science (but not both)”, but we are somewhat certain that it is computer science. This corresponds to the possibility distribution $\pi_{take(Sally, \cdot)}(math) = 1 - \alpha$; $\pi_{take(Sally, \cdot)}(CS) = \alpha$ if we are certain at level α that it is computer science. Similarly, we may want to express that it is α -certain that “Alice takes math or physics (but in any case not, both) if Sally takes physics”. This latter information translates in the possibilistic formulas

$$(take(Alice, math) \vee take(Alice, physics), \min(\alpha, [take(Sally, physics)]))$$

and $(\neg take(Alice, math) \vee \neg take(Alice, physics), 1)$.

Let us now formally define the concept of a possibilistic c -table.

Definition 3. A *possibilistic c -table* on X is a triple (T, Φ_T, \mathcal{P}) where T is any finite set of c -tuples on X , $\Phi_T \in \mathcal{S}$ is a global condition on T , and \mathcal{P} is a set of possibility distributions on some variables of $V = \bigcup_{A \in X} V(A)$.

In this definition, \mathcal{P} corresponds to the set of “soft constraints” (possibilistic restrictions) bearing on some variables of the c -table. First observe that the following property trivially holds.

Property 1. In the special case where possibility degrees take their values in $\{0, 1\}$, possibilistic c -tables reduce to regular c -tables.

Example 7. Let us consider the two following possibilistic c -tables (Table 6).

Table 7 represents the result of $q = \pi_{\{Student, Room\}}(r \bowtie_{Course=Course} s)$. The tuple $t = \langle Sally, Turing \rangle$ is a possible (resp. certain) answer to the degree 1 (resp. 0.4). Indeed, the most possible world such that the result of q contains t (resp. does not contain t) corresponds to the valuation $\{x = 0, y = Fermat\}$ (resp. $\{x = 1, y = Fermat\}$) whose possibility degree equals 1 (resp. 0.6). \diamond

Table 7. Result of the query from Example 7

<i>Student</i>	<i>Room</i>	
		$x = 0$ (1) \oplus $x = 1$ (0.6)
		$y = Fermat$ (1) \oplus $y = Turing$ (0.3)
<i>Sally</i>	<i>Turing</i>	$x = 0 \vee (x = 1 \wedge y = Turing)$
<i>Sally</i>	<i>Fermat</i>	$x = 1 \wedge y = Fermat$

Remark 2. It is important to emphasize that as soon as candidate values may be attached a degree of possibility, the worlds of $rep(T)$ become more or less possible (they would be more or less probable if a probabilistic database model were used). This has an impact on the notion of possible and certain answers (cf. Eqs. 2 and 3). In the graded possibilistic c -tables model we introduce, the degree of possibility (resp. certainty) associated with an answer t to a query q corresponds to the maximum of the possibility degrees attached to the worlds of $rep(\bar{q}(T))$ that contain t (resp. 1 minus the maximum of the possibility degrees attached to the worlds of $rep(\bar{q}(T))$ that do not contain t). \square

When a consequence of interest is of the form (p, α) , where α is a certainty level that depends on nothing, we have an α -certain answer. Besides, if we get only $(p, [q])$, and more generally $(p, \min(\alpha, [q]))$, and if $[q]$ is not known as being equal to 1, the answer p can be regarded as being *possible* at level 1. Indeed the possibility distribution associated with $(p, \min(\alpha, [q]))$ is such that $\pi_{\{(p, \min(\alpha, [q]))\}}(pq) = 1$; $\pi_{\{(p, \min(\alpha, [q]))\}}(p\bar{q}) = 1$; $\pi_{\{(p, \min(\alpha, [q]))\}}(\bar{p}q) = 1 - \alpha$ and $\pi_{\{(p, \min(\alpha, [q]))\}}(\bar{p}\bar{q}) = 1 - \alpha$, and thus $\Pi(p) = \max_{\omega \models p} \pi_{\{(p, \min(\alpha, [q]))\}}(\omega) = 1$. This indicates that the certainty level-based possibilistic logic cannot alone account for intermediary possibility levels, as further discussed now. What is computed here is only an upper bound of the possibility level, exploiting a part of the information only.

Example 8. Consider again the situation where Sally is taking math with possibility $1 - \alpha$ or computer science with possibility 1 (but not both, with full certainty), which writes in possibilistic logic $(take(Sally, CS), \alpha)$ and $(take(Sally, CS) \vee take(Sally, math), 1)$ (this latter formula acknowledges that Sally studies either CS or math). Let us evaluate the query “find the students who take math”, which translates into $(\neg take(x, math) \vee answer(x), 1)$; we obtain

$$(answer(Sally), [\neg take(Sally, CS)]),$$

but we do not retrieve $\pi_{take(Sally, \cdot)}(math) = 1 - \alpha$. This can be only done by applying the mixed resolution pattern recalled in Sect. 3, namely here

$$\begin{aligned} N(\neg take(x, math) \vee answer(x)) &= 1, \Pi(take(Sally, math)) = 1 - \alpha \\ &\models \Pi(answer(Sally)) \geq 1 - \alpha. \diamond \end{aligned}$$

However, although it does not appear on this very simple example, the evaluation of the possibility levels associated with the possibility distribution underlying

the uncertain and imprecise database may lead to complicated expressions and heavy computation, as in the probabilistic case.

Theorem 1. *Possibilistic c -tables are a strong representation system for relational algebra.*

Sketch of the Proof. We need to prove (cf. Formula 1) that: $rep(\bar{q}(T)) = q(rep(T))$. In the possibilistic c -tables model, an element of $rep(T)$ is a pair $(W_i, \Pi(W_i))$, where W_i is a possible world of T and $\Pi(W_i)$ is its associated possibility degree. Then, $q(rep(T))$ corresponds to the weighted set of worlds $\{\Pi(W_i)/q(W_i) \mid W_i \in rep(T)\}$ (remark: we keep the max of the possibility degrees in case of duplicate worlds). Let us denote by $\mathcal{W}(T')$ the possible worlds (without the associated possibility degrees) represented by the possibilistic c -table T' . Since regular c -tables are a strong representation system for relational algebra (see [18]), and since the definitions of the algebraic operators remain unchanged, we have $\mathcal{W}(rep(\bar{q}(T))) = \mathcal{W}(q(rep(T)))$ where $\mathcal{W}(q(rep(T))) = \{q(W_i) \mid W_i \in rep(T)\}$. Now, all one needs is a sound way to compute the possibility degree of a world generated by a possibilistic c -table (i.e. of a valuation that satisfies the conditions in the c -table). This way is provided by the axioms of possibility theory regarding conjunction and disjunction, and the computation is based on the possibility distributions attached to the possibilistic c -table on the one hand, and the conditions attached to the tuples on the other hand. \square

Let us now consider the counterparts of the yes-no queries discussed at the end of Sect. 2 and their associated complexity. In the possibilistic c -tables framework these queries are not of type yes-no anymore but are of the form “to which extent is it possible (resp. certain) that ...”. For instance, the containment problem $cont(q_0, q)$ now corresponds to the question: to which extent is it possible (resp. certain) that the result of a given query q_0 over a given set of c -tables \mathcal{S}_0 is included in the result of a given query q over a given set of c -tables \mathcal{S} ? Just as the complexity of possibilistic logic is the one of classical logic multiplied by the logarithm of the number of levels used in the scale [19], the complexity here remains in the same class as the one for regular c -tables.

4.3 The Particular Case of the Certainty-Based Model

In [22], we defined a model that we called “certainty-based” for representing relational databases containing uncertain attribute values, when some knowledge is available about the more or less certain value (or disjunction of values) that a given attribute in a tuple may *take*.

As the possibilistic model described in [6], the certainty-based model [22] relies on possibility theory [28]. However, it only keeps pieces of information that are more or less certain and leaves aside what is just possible. This corresponds to the most important part of information (a possibility distribution is “summarized” by keeping its most plausible elements, associated with a certainty level). For instance, $\langle 037, \text{John}, (40, \alpha) \rangle$ denotes the existence of a person

named John, whose age is 40 with certainty α . Then the possibility that his age differs from 40 is upper bounded by $1 - \alpha$ without further information.

The model can also deal with disjunctive uncertain values. For instance, $\langle 3, Peter, (Newton \vee Quincy, 0.8) \rangle$ represents the fact that it is 0.8-certain that the person number 3 named Peter lives in Newton or in Quincy. Then, the underlying possibility distributions π are of the form $\pi(u) = \max(A(u), 1 - \alpha)$ where A is an α -certain subset of the attribute domain and $A(u)$ equals 1 if $u \in A$, 0 otherwise.

Moreover, since some operations (e.g., the selection) may create “maybe tuples”, each tuple t from an uncertain relation r has to be associated with a degree N expressing the certainty that t exists in r . It will be denoted by N/t .

Example 9. Let us consider a relation r of schema $(\#id, Name, City)$ containing tuple $t_1 = \langle 1, John, (Quincy, 0.8) \rangle$, and the query “find the people who live in Quincy”. Let the domain of attribute *City* be $\{Boston, Newton, Quincy\}$. The answer contains $0.8/t_1$ since it is 0.8 certain that t_1 satisfies the requirement, while the result of the query “find the people who live in Boston, Newton or Quincy” contains $1/t_1$ since it is totally certain that t_1 satisfies the condition. \diamond

To sum up, a tuple $\alpha/\langle 037, John, (Quincy, \beta) \rangle$ from relation r means that it is α certain that person 037 exists in the relation, and that it is β certain that 037 lives in Quincy (independently from the fact that it is or not in relation r).

Obviously, this model is a particular case of a gradual possibilistic c -table where the only conditions present in a relation are used to represent the more or less certain value (or disjunction of values) that an attribute in a tuple may *take*. Of course, when using a c -table to represent such a relation, there is no need for the extra attribute N since the certainty level attached to a tuple is computed by evaluating the condition associated with this tuple (interpreting the conjunction as the minimum according to the axioms of possibility theory).

We have extended relational algebra in this context and shown that the model constitutes a representation system for this set of operators. The only constraints concern (i) the join that has to be based on an equality condition, (ii) the Cartesian product and join operations that must *take* independent relations as arguments. An important result is that the data complexity of these operations is the same as in the classical database case. This is also the case of general possibilistic c -tables when it comes to computing the “compact” result of a query, i.e., its resulting possibilistic c -table, but of course not when it comes to answering generalized yes-no queries (cf. the end of Subsect. 4.2). Since the certainty-based model does not include intertuple dependencies, a table that represents the result of a query in this model is easily interpretable. On the other hand, possibilistic c -tables are more expressive but also more complex and can only be exploited by an end-user through generalized yes-no queries.

Table 8. Probabilistic conditional table from Example 10

<i>Student</i>	<i>Course</i>	
		$x = \textit{math} (0.3) \oplus x = \textit{phys} (0.3) \oplus x = \textit{chem} (0.4)$ $t = 0 (0.15) \oplus t = 1 (0.85)$
<i>Alice</i>	<i>x</i>	
<i>Bob</i>	<i>x</i>	$(x = \textit{physics}) \oplus (x = \textit{chem})$
<i>Theo</i>	<i>math</i>	$t = 1$

4.4 Comparison with Probabilistic *c*-tables

We introduce probabilistic *c*-tables by means of the following example drawn from [17].

Example 10. Suppose Alice takes a course that is *math* with probability 0.3, or *physics* with probability 0.3, or *chem* with probability 0.4; Bob takes the same course as Alice provided that the course is *physics* or *chem*; Theo takes *math* with probability 0.85. This can be represented by the probabilistic *c*-table depicted in Table 8.

We may easily imagine a possibilistic version of the example. Suppose that Alice takes a course that is *math* with possibility α , or *physics* with possibility α , or *chem* with possibility 1. Bob takes the same course as Alice provided that the course is *physics* or *chem*. Theo takes *math* with possibility 1 and does not take *math* with possibility β . Now $D_{\textit{course}} = \{\textit{math}, \textit{chem}, \textit{physics}\}$. The pieces of information above can be expressed in possibilistic logic as:

$$\begin{aligned}
 &(\textit{take}(\textit{Alice}, \textit{chem}), 1 - \alpha) \\
 &(\textit{take}(\textit{Bob}, \textit{chem}), [\textit{take}(\textit{Alice}, \textit{chem})]) \\
 &(\textit{take}(\textit{Bob}, \textit{physics}), [\textit{take}(\textit{Alice}, \textit{physics})]) \\
 &(\textit{take}(\textit{Theo}, \textit{math}), 1 - \beta)
 \end{aligned}$$

From which one can deduce, e.g., $(\textit{take}(\textit{Bob}, \textit{chem}), 1 - \alpha)$. \diamond

Note that in case we would have

$$x = \textit{math} (0.25) \oplus x = \textit{phys} (0.35) \oplus x = \textit{chem} (0.4)$$

we would have to add

$$(\textit{take}(\textit{Alice}, \textit{chem}) \vee \textit{take}(\textit{Alice}, \textit{physics}), 1 - \alpha')$$

with $\alpha' < \alpha$ to $(\textit{take}(\textit{Alice}, \textit{chem}), 1 - \alpha)$.

Moreover, in case we would have

$$x = \textit{math} (0.4) \oplus x = \textit{phys} (0.4) \oplus x = \textit{chem} (0.2)$$

we would have to replace $(\textit{take}(\textit{Alice}, \textit{chem}), 1 - \alpha)$ by

$$(\textit{take}(\textit{Alice}, \textit{math}) \vee \textit{take}(\textit{Alice}, \textit{physics}), 1 - \alpha).$$

The above examples suggest that probabilistic *c*-tables and possibilistic *c*-tables are quite close as a representation tool, although obeying to different inference principles. Compared to probabilistic *c*-tables, an argument in favor of the possibilistic *c*-tables model lies in its robustness (i.e., in the fact that the order of

Table 9. Relations r (top) and s (bottom) — Possibilistic c -tables

#id	name	city
		$x = \text{Newton} (1) \oplus x = \text{Quincy} (0.2)$
		$y = \text{Quincy} (1) \oplus y = \text{Gardner} (0.6)$
		$z = \text{Newton} (1) \oplus z = \text{Gardner} (0.8) \oplus z = \text{Quincy} (0.7)$
		$u = \text{Quincy} (1) \oplus u = \text{Gardner} (1)$
37	John	x
53	Mary	y
72	Paul	z
81	Lisa	u

city	flea market
	$v = \text{yes} (1) \oplus v = \text{no} (0.3)$
	$w = \text{yes} (0.1) \oplus w = \text{no} (1)$
	$r = \text{yes} (1) \oplus r = \text{no} (0.4)$
Newton	v
Quincy	w
Gardner	r

the answers obtained is less sensitive to the values of the degrees in the distributions attached to the variables), which is illustrated by the toy example hereafter.

Example 11. Let us consider the possibilistic c -tables r of schema $(\#id, name, city)$, and s of schema $(city, flea\ market)$ describing respectively a set of people whose city of residence is ill-know, and a set of city for which we are not sure if they have a flea market or not. Let us consider the query q asking for the people who live in a city with a flea market: $\pi_{Name}(r \times_{city=city} (\sigma_{flea\ market=yes}(s)))$.

Let us first consider the possibilistic c -tables model (cf. Table 9). The resulting possibilistic c -table is represented in Table 10.

Table 10. Result of query q

name	
	$x = \text{Newton} (1) \oplus x = \text{Quincy} (0.2)$
	$y = \text{Quincy} (1) \oplus y = \text{Gardner} (0.6)$
	$z = \text{Newton} (1) \oplus z = \text{Gardner} (0.8) \oplus z = \text{Quincy} (0.7)$
	$u = \text{Quincy} (1) \oplus u = \text{Gardner} (1)$
	$v = \text{yes} (1) \oplus v = \text{no} (0.3)$
	$w = \text{yes} (0.1) \oplus w = \text{no} (1)$
	$r = \text{yes} (1) \oplus r = \text{no} (0.4)$
John	$(x = \text{Newton} \wedge v = \text{yes}) \vee (x = \text{Quincy} \wedge w = \text{yes})$
Mary	$(y = \text{Quincy} \wedge w = \text{yes}) \vee (y = \text{Gardner} \wedge r = \text{yes})$
Paul	$(z = \text{Newton} \wedge v = \text{yes}) \vee (z = \text{Gardner} \wedge r = \text{yes}) \vee (z = \text{Quincy} \wedge w = \text{yes})$
Lisa	$(u = \text{Quincy} \wedge w = \text{yes}) \vee (u = \text{Gardner} \wedge r = \text{yes})$

Table 11. Relations r (top) and s (bottom) — Probabilistic c -tables

<i>#id name</i>	<i>city</i>
	$x = \textit{Newton}$ (0.8) \oplus $x = \textit{Quincy}$ (0.2)
	$y = \textit{Quincy}$ (0.6) \oplus $y = \textit{Gardner}$ (0.4)
$z = \textit{Newton}$ (0.5) \oplus $z = \textit{Gardner}$ (0.3) \oplus $z = \textit{Quincy}$ (0.2)	
	$u = \textit{Quincy}$ (0.5) \oplus $u = \textit{Gardner}$ (0.5)
37 <i>John</i>	x
53 <i>Mary</i>	y
72 <i>Paul</i>	z
81 <i>Lisa</i>	u

<i>city</i>	<i>flea market</i>
$v = \textit{yes}$ (0.7) \oplus $v = \textit{no}$ (0.3)	
$w = \textit{yes}$ (0.1) \oplus $w = \textit{no}$ (0.9)	
$r = \textit{yes}$ (0.6) \oplus $r = \textit{no}$ (0.4)	
<i>Newton</i>	v
<i>Quincy</i>	w
<i>Gardner</i>	r

The answers obtained are thus:

John $\Pi = \max(\min(1, 1), \min(0.2, 0.1)) = 1,$
 $N = \max(\min(1 - 0.2, 1 - 0.3), \min(1 - 1, 1 - 1)) = 0.7$

Mary $\Pi = \max(\min(1, 0.1), \min(0.6, 1)) = 0.6,$
 $N = \max(\min(1 - 0.6, 1 - 1), \min(1 - 1, 1 - 0.4)) = 0$

Paul $\Pi = \max(\min(1, 1), \min(0.8, 1), \min(0.7, 0.1)) = 1,$
 $N = \max(\min(1 - 0.8, 1 - 0.3), \min(1 - 1, 1 - 0.4), \min(1 - 1, 1 - 1)) = 0.2$

Lisa $\Pi = \max(\min(1, 0.1), \min(1, 1)) = 1,$
 $N = \max(\min(1 - 1, 1 - 1), \min(1 - 1, 1 - 0.4)) = 0.$

John is a completely possible answer ($\Pi = 1$) since it is completely possible that (i) he lives in Newton, and (ii) Newton has a flea market. On the other hand, it is only 0.7 certain that he is an answer since it is 0.3 possible that Newton does not have a flea market. Since one has $N > 0 \Rightarrow \Pi = 1$, one may rank the answers in decreasing order of N first, then, for those such that $N = 0$, in decreasing order of Π . We get the following ranking: $\textit{John} \succ \textit{Paul} \succ \textit{Lisa} \succ \textit{Mary}$.

Finally, let us use a probabilistic model. In Table 11, the probability values are roughly specified, in agreement with the uncertainty ordering specified in the previous possibilistic tables. The result of q is the same as in the possibilistic case (cf. Table 10) except for the global conditions in which the degrees are different. Finally, we get the answers:

John $pr = 0.8 \times 0.7 + 0.2 \times 0.1 = 0.58$
Mary $pr = 0.6 \times 0.1 + 0.4 \times 0.6 = 0.3$
Paul $pr = 0.5 \times 0.7 + 0.3 \times 0.6 + 0.2 \times 0.1 = 0.55$
Lisa $pr = 0.5 \times 0.1 + 0.5 \times 0.6 = 0.35$

As can be seen, we again obtain the same ranking as in the possibilistic c -tables model. However, a rather slight modification of the probability values may lead to a modification of the ranking. For instance, if the probability distribution associated with Paul's city were changed into $\{0.55/Newton, 0.35/Gardner, 0.1/Quincy\}$, Paul would get the degree 0.605 and would be ranked first (before John). This contrasts with the possibilistic situation where the result remains stable as long as the ordering of the possibilistic values is not changed. \diamond

5 Conclusion

Possibility theory and c -tables have appeared at about the same time, at the end of the 1970's, in two different areas of information processing. Curiously, they had remained unrelated until now. This paper provides a first study in order to bridge them. Indeed, c -tables, as a convenient way of describing possible worlds, can be easily extended to a possibilistic modeling of uncertain and imprecise information. This provides a general setting that appears quite appropriate for handling uncertainty in a qualitative way. The qualitative nature of possibility theory makes simpler the elicitation of the possibility and certainty degrees, and leads to a modeling less sensitive to modifications of the values of the degrees than in a probabilistic framework. Moreover, the particular case of the certainty-based approach is especially tractable. Besides, the existing relation between answer-set programming and generalized possibilistic logic [13] and the underlying role of possibilistic logic with respect to possibilistic c -tables suggests to study a possibilistic version of Datalog [2, 14] in the future.

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading (1995)
2. Abiteboul, S., Kanellakis, P.C., Grahne, G.: On the representation and querying of sets of possible worlds. *Theor. Comput. Sci.* **78**(1), 158–187 (1991)
3. Antova, L., Jansen, T., Koch, C., Olteanu, D.: Fast and simple processing of uncertain data. In: Proceedings of the 24th International Conference on Data Engineering (ICDE 2008), pp. 983–992 (2008)
4. Benjelloun, O., Das Sarma, A., Halevy, A., Theobald, M., Widom, J.: Databases with uncertainty and lineage. *VLDB J.* **17**(2), 243–264 (2008)
5. Benjelloun, O., Das Sarma, A., Halevy, A., Widom, J.: ULDBs: databases with uncertainty and lineage. In: Proceedings of VLDB 2006, pp. 953–964 (2006)
6. Bosc, P., Pivert, O.: About projection-selection-join queries addressed to possibilistic relational databases. *IEEE T. Fuzzy Syst.* **13**(1), 124–139 (2005)
7. Codd, E.F.: Extending the relational database model to capture more meaning. *ACM Trans. Database Syst.* **4**(4), 397–434 (1979)
8. Dalvi, N., Suciu, D.: Management of probabilistic data: foundations and challenges. In: Proceedings of PODS 2007, pp. 1–12 (2007)
9. Dubois, D., Lang, J., Prade, H.: Possibilistic logic. In: Gabbay, D.M., Hogger, C.J., Robinson, J.A., Nute, D. (eds.) *Handbook of Logic in Artificial Intelligence and Logic Programming*, vol. 3, pp. 439–513. Oxford University Press, Oxford (1994)

10. Dubois, D., Prade, H.: *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York (1988). (with the collaboration of H. Farreny, R. Martin-Clouaire, and C. Testemale)
11. Dubois, D., Prade, H.: Possibility theory: qualitative and quantitative aspects. In: Gabbay, D.M., Smets, P. (eds.) *Quantified Representation of Uncertainty and Imprecision. Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 1, pp. 169–226. Kluwer Academic Publishers, Dordrecht (1998)
12. Dubois, D., Prade, H.: Possibilistic logic – an overview. In: Siekmann, J.H. (ed.) *Computational Logic. Handbook of the History of Logic*, vol. 9, pp. 283–342. Elsevier, Amsterdam (2014)
13. Dubois, D., Prade, H., Schockaert, S.: Stable models in generalized possibilistic logic. In: Brewka, G., Eiter, T., McIlraith, S.A. (eds.) *Proceedings of the 13th International Conference on Principles of Knowledge Representation and Reasoning (KR 2012)*, pp. 519–529. AAAI Press, Menlo Park (2012)
14. Gallaire, H., Minker, J.J. (eds.): *Advances in Data Base Theory*. In: *Proceedings of the Symposium on Logic and Data Bases, Centre d' Etudes et de Recherches de Toulouse, 1977*. Plenum Press (1978)
15. Grahne, G.: *The Problem of Incomplete Information in Relational Databases*. LNCS, vol. 554. Springer, Heidelberg (1991)
16. Green, C.: Theorem-proving by resolution as a basis for question-answering systems. In: Michie, D., Meltzer, B. (eds.) *Machine Intelligence*, vol. 4, pp. 183–205. Edinburgh University Press, Edinburgh (1969)
17. Green, T.J., Tannen, V.: Models for incomplete and probabilistic information. In: *Proceedings of the IIDB 2006 Workshop*, pp. 278–296 (2006)
18. Imielinski, T., Lipski, W.: Incomplete information in relational databases. *J. ACM* **31**(4), 761–791 (1984)
19. Lang, J.: Possibilistic logic: complexity and algorithms. In: Kohlas, J., Moral, S. (eds.) *Algorithms for Uncertainty and Defeasible Reasoning. Handbook of Defeasible Reasoning and Uncertainty Management Systems*, vol. 5, pp. 179–220. Kluwer Academic Publisher, Dordrecht (2001). (Gabbay, D.M. and Smets, Ph., eds.)
20. Lipski, W.: On semantic issues connected with incomplete information databases. *ACM Trans. Database Syst.* **4**(3), 262–296 (1979)
21. Lipski, W.: On databases with incomplete information. *J. ACM* **28**, 41–70 (1981)
22. Pivert, O., Prade, H.: A certainty-based model for uncertain databases. *IEEE Trans. Fuzzy Syst.* **23**(4), 1181–1196 (2015)
23. Prade, H.: The connection between Lipski's approach to incomplete information data bases and Zadeh's possibility theory. In: *Proceedings of the International Conference on Systems Methodology, Washington, D.C., 5–9 January*, pp. 402–408 (1982)
24. Prade, H.: Lipski's approach to incomplete information data bases restated and generalized in the setting of Zadeh's possibility theory. *Inf. Syst.* **9**(1), 27–42 (1984)
25. Prade, H., Testemale, C.: Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries. *Inf. Sci.* **34**(2), 115–143 (1984)
26. Shimizu, S., Ishihara, Y., Takarabe, T., Ito, M.: A probabilistic database model with representability of dependency among tuples. In: *Proceedings of the 4th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2000)*, pp. 221–225 (2000)
27. Suciuc, D., Olteanu, D., Ré, C., Koch, C.: *Probabilistic Databases. Synthesis Lectures on Data Management*. Morgan & Claypool Publishers (2011)
28. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.* **1**(1), 3–28 (1978)