Querying Possibilistic Databases: Three Interpretations

Patrick Bosc and Olivier Pivert

IRISA–ENSSAT, Lannion, France
{bosc,pivert}@enssat.fr

1 Introduction

Many authors have made proposals to model and handle databases involving uncertain data. In particular, the last two decades have witnessed a blossoming of researches on this topic (cf. e.g., [3,4,19] for some recent ones). Even though most of the literature about uncertain databases uses probability theory as the underlying uncertainty model, some approaches rather rest on possibility theory [26]. The initial idea consisting in applying possibility theory to this issue goes back to the early 80's [24]. More recent advances on this topic can be found in [10]. 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 attached to candidate values;
- 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.

Our aim is not to claim (nor to demonstrate) that the possibility-theory-based framework is "better" than the probabilistic one at modeling uncertain databases, but that it constitutes an interesting alternative inasmuch as it captures a different kind of uncertainty (of a qualitative nature). An example is that of a person who witnesses a car accident and is not sure about the model of the car involved. In such a case, it seems reasonable to model the uncertain value by means of a possibility distribution, e.g., {1/Mazda, 1/Toyota, 0.7/Honda} — where 0.7 is a numerical encoding in a usually finite possibility scale — rather than with a probability distribution which would be artificially normalized.

The rest of the paper is organized as follows. Section 2 is devoted to a reminder about basic notions concerning the interpretation of an uncertain database in terms of a set of possible worlds. In Section 3, two models of uncertain databases founded on possibility theory are presented. Then, in Section 4, three fairly different families of queries are proposed, that have quite different meanings. Section 5 concludes the paper and opens some lines for future works.

2 Basic Notions

2.1 The Possible Worlds Semantics

The possible worlds model is founded on the fact that uncertainty in data makes it impossible to define what precisely the real world is. One can only describe the set of

R.R. Yager et al. (Eds.): *Soft Computing: State of the Art Theory*, STUDFUZZ 291, pp. 161–176.
DOI: 10.1007/978-3-642-34922-5_12

possible worlds which are consistent with the available information. As far as a table T conveys some imprecision/uncertainty, several interpretations (I) can be drawn from T and the set of all the interpretations of T is denoted by $rep(T)$. The notation $rep(D)$ extends naturally to an uncertain database D involving several tables. A regular database is nothing but a special case of an uncertain one which has only one interpretation. From a semantic point of view, such an uncertain database D can be interpreted in terms of a set of usual databases, also called worlds W_1, ..., W_p, and $rep(D) = \{W_1, ..., W_p\}$. In the following, we consider the case where $rep(D)$ is finite. Any world W_i is obtained by choosing a candidate value in each set appearing in a relation T_j pertaining to D. One of these (regular) databases, let us say W_k, is supposed to correspond to the actual state of the universe modeled. The assumption of independence between the sets of candidates is usually made and then any world W_i corresponds to a conjunction of independent choices (thus the degree associated to a world is based on a conjunction operator, e.g., "min" or "product").

Example 1. Let us consider the uncertain database D involving a single relation *im* whose schema is *IM*(#*i*, *airc*, *date*, *place*). Relation *im* is assumed to describe satellite images of aircrafts. Each image, numbered (#*i*), was taken on a certain location (*place*) a given day (*date*) and it is supposed that it includes a single aircraft (*airc*). With the extension of *im* depicted in Table 1 six worlds can be drawn, W_1, W_2, W_3, W_4, W_5 and W_6 since there are three candidates for *date* in the first tuple and two candidates for *airc* in the second one. Two of the worlds associated with the uncertain relation *im* are represented in Table 1. ⋄

Table 1. An extension of *im* (top) and two worlds associated with it (bottom)

#*i*	*airc*	*date*	*place*
i_1	a_1	$\{d_1, d_3, d_7\}$	c_1
i_3	$\{a_3, a_4\}$	d_1	c_2

#*i*	*airc*	*date*	*place*
i_1	a_1	d_1	c_1
i_3	a_3	d_1	c_2

#*i*	*airc*	*date*	*place*
i_1	a_1	d_7	c_1
i_3	a_4	d_1	c_2

2.2 Strong Representation Systems and Compact Calculus

When dealing with an uncertain database D, a very important issue is that of the efficiency of the querying process. A naive way of doing would be to make explicit all the interpretations of D (at least when they are finite) in order to query each of them. Such an approach is intractable in practice and it is of prime importance to find a more realistic alternative. To this end, the notion of a representation system has been introduced — initially by Imielinski and Lipski [22] — and discussed in [1]. The basic idea is to look for a way for representing both initial tables and those resulting from queries so that the representation of the result of a query q against any database D (made of tables T_1, ..., T_p) denoted by $q(D)$, is equivalent (in terms of interpretations, or worlds) to the set of results obtained by applying q to every interpretation of D, i.e.:

$$rep(q(D)) = q(rep(D)) \tag{P1}$$

where $q(rep(D)) = \{q(W) \mid W \in rep(D)\}$. If property P1 holds for a representation system ρ and a subset σ of the relational algebra, ρ is called a *strong representation system* for σ. From a querying point of view, P1 enables a direct (or compact) calculus of a query q, which then applies to D itself without making the worlds explicit (see Figure 1). So doing, provided that relational operations are defined over tables of the system considered, reasonable performances can be expected.

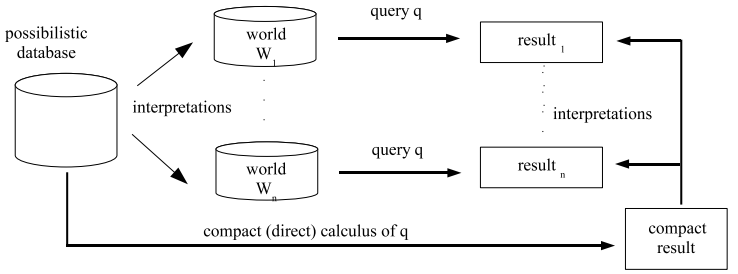

Fig. 1. Compact query evaluation

3 Two Uncertain Database Models Based on Possibility Theory

3.1 Full Possibilistic Model

In the "full possibilistic model" [10], any attribute value can be a possibility distribution which acts as a restriction over the values that are more or less preferred for a considered attribute (a precise value is an extreme case where only one candidate is possible). Besides, there is a need for expressing that some tuples may not be represented in some worlds. Indeed, a selection may lead to discard candidate values from a distribution, but one must be able to compute the degree of any world of the answer, including those in which some tuples are not represented. A simple solution is to introduce a new attribute, denoted by N, which states whether or not it is legal to build worlds where the corresponding tuple has no representative, and, if so, the influence of this choice in terms of possibility degree. N expresses the certainty of the presence of a representative of the tuple in any world. By doing so, it is possible to generate the worlds in which a tuple is not represented, by taking into account the degree of possibility of its absence, which, according to possibility theory, is given by $(1 - N)$. A tuple is denoted as a pair N/t where N equals 1 for tuples of initial possibilistic relations as well as when no alternative has been discarded. A second aspect is related to the fact that it is sometimes necessary to express dependencies between candidate values of different attributes of a same tuple. For instance, let A and B be two attributes whose respective candidates in a given tuple t are $\{a_1, a_2\}$ and $\{b_1, b_2, b_3\}$. If, according to a given selection criterion, the only legal associations are (a_1, b_1) and (a_2, b_3), one cannot call on a Cartesian

product of subsets of $t.A$ and $t.B$. In other words, A and B values cannot be kept separate (which would mean that they are independent) and the correct associations can be explicitly represented if the model incorporates attribute values defined as possibility distributions over several domains. In this context, candidates can be (weighted) tuples in a model based on the concept of nested relations. Besides, let us emphasize two particular aspects, both connected with the fact that nested relations are used to support possibility distributions: i) tuples of nested relations are weighted since any element of a possibility dis-tribution is assigned a level of preference and ii) the extension of a nested relation has a disjunctive meaning according to the semantics of a possibility distribution. The notation

$$R(A_1, \ldots, A_m, X_1(A_p, \ldots, A_q), \ldots, X_n(A_k, \ldots, A_r))$$

stands for a schema in which A_1 to A_m are elementary attributes (also called level-one attributes) whose values are either precise or possibility distributions and $X_i(A_h, \ldots, A_j)$ represents a "structured" attribute X_i whose values are possibility distributions made of tuples built over attributes A_h to A_j which are called "nested" attributes. Obviously, such relations have an interpretation in terms of worlds as it is the case for ordinary possibilistic relations. When one moves to a given world, a structured candidate value is split into atomic values and the schema becomes unnested. The idea is to use the extended model to represent the result of intermediate operations in a correct fashion.

Table 2. An extension of relation r

		X				
A	B	C	D	E	π	N
a_1	$\{\pi_1/b_1, \pi_2/b_4\}$	c_2	d_1	e_3	π_3	1
a_2	b_3	c_1	d_1	e_2	π_4	0.4
		c_3	d_2	e_3	π_5	
		c_2	d_4	e_2	π_6	
		c_2	d_1	e_3	π_7	

Example 2. Let us consider the intermediate relation of schema $R(A, B, X(C, D, E))$ represented by Table 2 where the π_i's denote possibility degrees. Five possibilities exist as to the second tuple since it may be absent ($N < 1$). Consequently, ten worlds can be derived from this imprecise relation. The world containing only the tuple $\langle a_1, b_4, c_2, d_1, e_3\rangle$, in which the second tuple is not represented, is associated with the degree:

$$min(min(1, \pi_2, \pi_3), 1 - 0.4).$$

The world with the two tuples $\langle a_1, b_1, c_2, d_1, e_3\rangle$ and $\langle a_2, b_3, c_3, d_2, e_3\rangle$ can also be drawn and its degree is:

$$min(min(1, \pi_1, \pi_3), min(1, 1, \pi_5)).\diamond$$

3.2 Certainty-Based Model

3.2.1 Main Features of the Model

In the certainty-based model [13,15], a possibility distribution is "synthetized" by keeping only its most plausible elements. So, to each uncertain value a of an attribute A is attached a certainty degree α. The underlying possibility distribution associated with an uncertain attribute value (a, α) is $\{1/a, (1-\alpha)/\omega\}$ where ω denotes $domain(A)\backslash\{a\}$ (due to the duality necessity/possibility: $N(a) \geq \alpha \Leftrightarrow \Pi(\omega) \leq 1-\alpha$ [21]). For instance, let us assume that the domain of attribute City is $\{Newton, Quincy, Boston\}$. The uncertain attribute value $(Newton, \alpha)$ is assumed to correspond to the possibility distribution $\{1/Newton, (1-\alpha)/Quincy, (1-\alpha)/Boston\}$. More generally, the model can deal with disjunctive values, and the underlying possibility distributions are of the form $\{max(\mu_S(x_1), 1-\alpha)/x_1, \ldots, max(\mu_S(x_p), 1-\alpha)/x_p\}$ where S is an α-certain subset of the attribute domain and $\mu_S(x_i)$ equals 1 if $x_i \in S$, 0 otherwise [20]. Let us notice that, in general, there is not a strict equivalence between an initial possibility distribution (e.g., $\{1/Newton, 1/Malden, 0.6/Quincy, 0.2/Boston\}$ and the distribution ($\{1/Newton, 1/Malden, 0.6/Quincy, 0.6/Boston\}$) derived from its synthetized form $(Newton \vee Malden, 0.4)$.

Moreover, since some operations may create "maybe tuples" (e.g., the selection as in the full possibilistic model), each tuple t from an imprecise 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 3. Let us consider the relation r of schema (#id, Name, City) containing tuple $t_1 = \langle 1, John, (Boston, 0.8)\rangle$, and the query "find the persons who live in Boston". Let the domain of attribute City be $\{Newton, Quincy, Boston\}$. 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 persons 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 $\alpha/\langle 37, John, (Boston, \beta)\rangle$ from relation r means that it is α certain that person 37 exists in the relation, that it is totally sure that the name of that person is *John*, and that it is β certain that 37 lives in *Boston* (independently from the fact that it is or not in relation r).

Given a query, only answers that are *somewhat certain* are considered of interest (in contrast with those that are just possible), which makes the approach much simpler. Consider the relations r and s from Table 3 and a query asking for the persons who live in a city with a flea market. *John* will be retrieved with a certainty level equal to $min(\alpha, \beta)$ (in agreement with the calculus of necessity measures [20]). Although it is not impossible that *Mary* lives in a city with a flea market, she does not belong to the answer because this is just possible.

As mentioned above, it is also possible to handle cases of disjunctive information in this setting. For instance, $\langle 3, Peter, (Gardner \vee Fitchburg, 0.8)\rangle$ represents the fact that it is 0.8-certain that the person number 3 named Peter lives in Gardner or in Fitchburg.

Table 3. Relations r (left) and s (right)

#id	*Name*	*City*	*N*
1	John	(Newton, α)	1
2	Mary	(Norwood, δ)	1

City	*Flea Market*	*N*
Newton	(yes, β)	1
Norwood	(no, γ)	1

3.2.2 Strong Representation System

Let us now examine what becomes of property P1 in such a context. Let us denote by D an imprecise database involving certainty levels, $poss(D)$ the corresponding imprecise database involving the simplified possibility distributions of Subsection 3.2.1 (i.e., those associated with values that are somewhat certain), q an algebraic query, and q_c the compact version of q. The counterpart of property P1 is:

$$q_c(D) = \psi(q(rep(poss(D)))) \tag{P2}$$

where $\psi(r')$ denotes the certainty-based relation which gathers the tuples somewhat certainly in the intersection of all the (more or less) possible worlds from the set r' (each world from r' represents a possible result of q applied to D).

Table 4. Extension of *im* for Example 4

#i	*airc*	*date*	*place*
7	{1/MiG31, 0.8/MiG29}	96/03/02	$\{1/v_1, 0.2/v_2\}$
9	{1/Su27, 0.3/Su30, 0.5/MiG31}	92/12/01	v_1
17	MiG31	96/09/27	$\{1/v_2, 0.4/v_1\}$
5	{1/MiG29, 1/Su7}	95/06/09	v_2
34	MiG31	95/10/01	v_1

Table 5. Result of the query of Example 4

#i	*airc*	*date*	*place*	Π	N
7	{1/MiG31, 0.8/MiG29}	96/03/02	$\{1/v_1, 0.2/v_2\}$	1	0.2
9	{1/Su27, 0.3/Su30, 0.5/MiG31}	92/12/01	v_1	0.5	0
17	MiG31	96/09/27	$\{1/v_2, 0.4/v_1\}$	0.4	0
34	MiG31	95/10/01	v_1	1	1

4 Three Families of Query Semantics

Though it would make sense to envisage fuzzy queries (i.e., involving preferences expressed through fuzzy predicates), for space reasons, we only focus on Boolean queries.

4.1 Event-Oriented Querying

The corresponding model and query language were first introduced in [24] where it was possible to issue fuzzy queries against a possibilistic database. First, it is important

to notice that this approach is not related to the possible worlds semantics. The idea is rather to see a query as a way to build facts (or events) as tuples using algebraic operations. Each tuple is assigned a pair of grades Π, N expressing the possibility and necessity of the corresponding event. The central operator is the selection for which ouput tuples are input tuples (kept unchanged) accompanied by the two grades Π and N mentioned before. In the presence of a Boolean selection condition ϕ applying to attribute A, the value of Π for tuple t (inside which A is represented as the possibility distribution $\pi_{t.A}$) is defind as:

$$sup_{d \in domainA}\, min(\pi_{t.A}(d), \phi(d)).$$

It equals 1 if there is (at least) one value in the core of $\pi_{t.A}$ that satisfies ϕ and 0 if no value of the support of $\pi_{t.A}$ matches ϕ. Of course, other values of the unit interval can be taken (see Example 4). Similarly, the necessity degree is given by:

$$1 - sup_{d \in domain(A)}\, min(\pi_{t.A}(d), \neg\phi(d)) = inf_{d \in domain(A)}\, max(1 - \pi_{t.A}(d), \phi(d)).$$

It equals 1 if any somewhat possible value of $\pi_{t.A}$ satisfies ϕ and 0 if a completely possible value of $\pi_{t.A}$ does not comply with ϕ. Of course, one has the property: $\Pi < 1 \Rightarrow N = 0$, as illustrated in the next example.

Example 4. Let us consider the relation *im* whose schema is given in Example 1 with the extension of Table 4. The query looking for images of "*MiG31*" taken in city v_1 returns the relation of Table 5.$\diamond$

It is worth noticing that, in such an approach, the composition of operations is problematic since input tuples are not "updated". For instance, the query looking for persons whose age is between 28 and 32 would reject $\langle John, \{1/25, 1/35\}\rangle$ whereas this tuple is selected if two successive selections are used.

4.2 Possible Worlds

4.2.1 Queries in the Full Possibilistic Model

Let us first point out some difficulties raised by the presence of disjunctive values. Let us consider the following relations $r(A, B)$ and $s(B, C)$:

$$r = \{\langle\{\alpha/a_1, \beta/a_2, \gamma/a_3\}, b\rangle\};\ s = \{\langle b, c_1\rangle, \langle b, c_2\rangle\}$$

where incompleteness is only due to the fact that the actual value of A in the tuple of r is either a_1, or a_2, or a_3. The natural join of r and s leads to a relation $t(A, B, C)$ involving two tuples, but it is mandatory to guarantee that only three possible worlds can be drawn from t (and not 3^2), since attribute A should take the same value in each of the two tuples, for property P1 to hold. Now, let us perform the natural join of the following relations:

$$r = \{\langle a, \{\alpha/b_1, \beta/b_2, \gamma/b_3\}\rangle\} \text{ and } s = \{\langle b_1, c_1\rangle, \langle b_3, \{\eta/c_2, \delta/c_3\}\rangle\}.$$

Here, the resulting relation is either empty, or made of a single tuple among three possible: $\langle a, b_1, c_1\rangle$, $\langle a, b_3, c_2\rangle$ and $\langle a, b_3, c_3\rangle$. It is then necessary to express that these four

situations are exclusive. This implies using a sophisticated data model such as c-tables introduced by Imielinski and Lipski [22], which in turn raises important complexity issues.

Binary relational operations may be categorized the following way. Type 1 (resp. type 2) operations are such that any tuple from an operand relation can take part in the generation of at most one (resp. several) tuple(s) in the resulting relation. An example of a type 1 operation is the union. Type 2 operations include the intersection, the difference, the Cartesian product and the join (in their most general forms). However, in some particular cases linked to the presence of keys, an operation that is in general of type 2 can behave as a type 1 one (for instance the join operation when the join attributes are precise and constitute the keys of the operand relations or the foreign-key join detailed later). To summarize, let us say that in a strict relational framework, it is not possible to define a strong representation system allowing to deal with an operation of type 2 in the presence of imprecise values [9].

We now give an overview of four operators which define a language for which the full possibilistic model is an SRS. The reader will find more details and examples in [9] and [10]. In the following, because of space limits, we consider the case where input relations only include level-one attributes.

Selection

The usual selection keeps the tuples of a relation which satisfy a given predicate. Here, the idea is to retain only candidate values complying with the selection criterion. We review the various cases of selection conditions and examine their impact on the structure of the result.

When the condition is of the form "*att* θ *constant*" ($\theta \in \{=, \neq, >, <, \geq, \leq\}$), the structure of the result is the same as that of the input relation. If the schema of the input relation r is $R(A, B)$, the condition concerns attribute A and $scv(t.A)$ denotes the non-weighted set of candidate values appearing in $t.A$, the selection is defined as:

$$select(r, \theta(A, v)) = \{N'/\langle restrict(t.A, \theta(A,v)), t.B\rangle \,|\, N/t \in r \wedge N' = min(N, 1 - sup_{x \in scv(t.A) \,|\, \neg\theta(x,v)} \pi_{t.A}(x))\}$$

with

$$restrict(t.A, \theta(A,v)) = \{\ldots + \pi/a + \ldots\} \text{ s.t. } a \in scv(t.A) \wedge \theta(a,v) \wedge \pi = \pi_{t.A}(a).$$

This formula says that, in any tuple t, only the elements of the distribution $t.A$ which satisfy the condition are retained in the resulting tuple. Moreover, the degree of certainty associated with this tuple ($t.N$) is updated according to the highest possible value which is discarded. It is proven in [9] that property P1 holds with this definition.

Let us now consider with selection conditions of the form "$A_1 \theta A_2$" or "$cond_1(A_1)$ or $cond_2(A_2)$". In both cases, if A_1 and A_2 are imprecise attributes, it is necessary to gather their candidate values in a nested relation so that only the correct pairs of values are kept in the result. The corresponding definition is given in [9]. The way the operator works is illustrated in the following example by a condition involving a disjunction.

Example 5. Let us consider the schema (#c, name, city, mileage) of an intermediate relation *ac* describing cars with their number, name, city of last owner and mileage. The condition (brand = "C*" or city = "Paris") applied to:

$$\{0.7/\langle 1, \{1/Camry, 0.4/Taurus\}, \{1/Madrid, 0.7/Paris\}, 75000\rangle\}$$

leads to the result:

$$\{0.6/\langle 1, \{1/\{Camry, Madrid\},\ 0.7/\{Camry, Paris\},\ 0.4/\{Taurus, Paris\}\}, 75000\rangle\}.$$

The necessity degree 0.6 attached to the tuple corresponds to $min(0.7, 1-\rho)$ where $\rho = 0.4$ is the possibility degree of the most possible pair of candidates that does not satisfy the selection criterion, i.e., $\langle Taurus, Madrid\rangle$ here. This way of doing guarantees the validity of property P1. ◇

Other Operators

As stated before, the classical join cannot apply in general for possibilistic relations due to the disjunctive nature of possibility distributions. However, we point out a specific type of join, called fk-join [9], where this problem does not appear since the tuples resulting from the join are independent in terms of their interpretation.

The operation $fk\text{-}join(r, s, (U, V))$ composes a possibilistic relation r whose schema is $R(U, Y)$ with a regular relation s (whose schema is $S(V, Z)$ where V is compatible with U) describing the graph of the functional dependency $V \rightarrow Z$. The fk-join computes the image of any imprecise U-value present in r by means of the function. In order to keep the elementary associations between antecedents and images of the FD $V \rightarrow Z$, it is mandatory to place U and Z candidate values inside a same nested relation. Let us consider the case where the schema of r is $R(A, B, G)$ with $U = \{A, B\}$ and the schema of s is $S(C, D, E)$ with $V = \{C, D\}$. The schema of the result is $Res(X(A, B, E), G)$.

Contrary to the usual case, the projection of a possibilistic relation does not entail any duplicate removal. One proceeds so that it is impossible to get a world after projection which would be more possible than the corresponding one before projection. This means that, for a given tuple, the possibility of the most possible candidate of the attributes which are removed becomes the upper bound of any interpretation of the tuple issued from the projection [7].

It is also possible to show that this model constitutes a strong representation system for the union operator provided that input relations are independent. Under this assumption, the union gathers the tuples of the two input relations and produces a result where the tuples are independent.

About Generalized Yes-No Queries

Queries addressed to an imprecise database may raise the problem of the interpretability of their results by an end-user. Indeed, even when "simple" models based on relations

without any conditions are used — such as that presented above —, it appears difficult for an end-user to grasp the content of a relation that may include nested subrelations, distributions of possible values and necessity degrees. This is why several authors have considered a class of queries which are more specialized (or targeted) to fit user needs. This is the case, for instance, of S. Abiteboul who studied such queries in the context of Codd-tables and tables with conditions [2]. In the context of possibilistic databases, the queries considered in [11,12] are basically yes-no questions about some properties possessed (or not) by some of the worlds of an imprecise database. Their general query format is: "to what extent is it possible {and/or} certain that the answer to q fulfills condition C?" where q is a (constrained) relational algebraic query which may include only the operators for which the model is an SRS, i.e. projection, selection, fk-join and union (cf. above). More precisely, the following types of queries are considered:

- vacuity-based yes-no queries: to what extent is it possible and certain that the answer to q is non-empty?
- tuple-membership-based yes-no queries: to what extent is it possible and certain that tuple t belongs to the answer to q?
- cardinality-based yes-no queries: to what extent is it possible and certain that the answer to q contains at least (resp. at most, exactly) k items?
- inclusion-based yes-no queries: to what extent is it possible and certain that the answer to q contains the set of tuples $\{t_1, ..., t_k\}$?

For each of these queries, the authors show that the processing obeys the following three step scheme:

1. pre-processing in order to eliminate the unnecessary attributes (and, for tuple-membership-based queries, to remove from the relations the tuples that cannot generate the target tuple);
2. evaluation of q, which yields a resulting possibilistic relation *res*;
3. post-processing aimed at computing the final possibility and certainty degrees Π and N.

The four previous types of queries can be clustered into two categories: those which require only a sequential scan of the result of q (vacuity and tuple-membership-based queries) and those for which it is necessary to use a "trial and error" type of algorithm (cardinality and inclusion-based queries).

4.2.2 Queries in the Certainty-Based Model

We now outline the compact version of the relational algebraic operators in the certainty-based database model [13,15]. The only limitation with respect to the usual algebraic framework consists in the fact that the operands of union, Cartesian product and join must be independent relations. Indeed, the presence of non-independent relations (for instance stemming from two selections on the same relation or a self join) might induce dependencies between uncertain values in a same tuple of the result, which cannot be handled in the model.

Selection

Let us consider a relation r of schema (A, X) where A is an attribute and X is a set of attributes, and a selection condition ϕ on A. Let us denote by $scv(t.A)$ the disjunctive set of values — which may be a singleton — somewhat certain for attribute A in tuple μ/t, and by $cl(t.A)$ the associated certainty level. Let us first deal with the case where ϕ writes $A\,\theta\,v$ where θ denotes a comparator and v a constant.

$$\begin{aligned} select(r, A\,\theta\,v) = \{\mu'/t \,|\, \exists \mu/t \in r \text{ s.t. } \forall a_i \in scv(t.A),\, a_i\,\theta\,q \wedge \\ \mu' = min(\mu, 1) = \mu \text{ if } \forall a_i \in domain(A),\, a_i\,\theta\,v; \\ \mu' = min(\mu, cl(t.A)) \text{ otherwise}\}. \end{aligned}$$

The proof that this definition of the selection satisfies property P2 can be found in [13]. The case of a condition ϕ of the form $A_1\,\theta\,A_2$ where A_1 and A_2 denote two attributes is dealt with in [13] but is omitted here for space reasons.

Example 6. Let us consider the database D made of the sole relation *emp* of schema (#id, name, city, job). Let us suppose that *emp* only contains tuple t = 0.9/⟨17, *John*, (*Boston*, 0.8), (*Engineer*, 0.7)⟩ and let us consider the query:

q = select(emp, city = 'Paris' and job = 'Engineer').

Its compact result is 0.7/⟨17, *John*, (*Boston*, 0.8), (*Engineer*, 0.7)⟩. Let us show that property P2 is satisfied. Identifier 17 is present in every completely possible world of the result. The most possible world of *emp* where 17 is not present in the result of the selection is made of the tuple ⟨17, *John*, *Boston*, ε)⟩ (where $\varepsilon \in \omega = domain(job)\backslash\{Engineer\}$) and has the possibility degree $min(1, 1 - 0.7) = 0.3$. Hence, the certainty degree attached to 17 in the result is 1 – 0.3 = 0.7. The most possible world where 17 has a *city* value different from *Boston* in the result has the possibility degree $1 - 0.8 = 0.2$. Hence, the certainty degree attached to the *city* value *Boston* in the tuple identified by 17 in the result is $1 - 0.2 = 0.8$. The most possible world where 17 has a *job* value different from *Engineer* in the result has the possibility degree $1 - 0.7 = 0.3$. Hence the certainty degree attached to the *job* value *Engineer* in the tuple identified by 17 in the result is $1 - 0.3 = 0.7$. The compact calculus is thus correct. ◇

Join

The compact definition of the join is:

$$\begin{aligned} join(r_1, r_2, A = B) = \{min(\alpha, \beta, \chi, \delta)/t_1 \oplus t_2 \,|\, \exists \alpha/t_1 \in r_1, \exists \beta/t_2 \in r_2 \text{ s.t.} \\ card(scv(t_1.A)) = 1 \wedge card(scv(t_2.B)) = 1 \wedge \\ scv(t_1.A) = scv(t_2.A) \wedge cl(t_1.A) = \chi \wedge cl(t_2.B) = \delta\} \end{aligned}$$

where $\oplus$ denotes the concatenation and *card* returns the cardinality of a set. Notice that only the tuples whose value for the join attribute is non-disjunctive (i.e., is a singleton) can participate in the result: for the other ones, one cannot be certain at all that they

match a tuple from the other relation. Indeed, for a tuple t_1 of r whose join attribute value $t_1.A$ is disjunctive, it is always possible to find a completely possible interpretation such that the (equi-)join condition is false, whatever the tuple t_2 from s. Note that property P2 would not hold in the case of a θ-join where θ is not equality. In [13], it is shown that the usual equivalence between a semi-join and a join followed by a projection: $r_1 \ltimes r_2 \equiv (r_1 \bowtie r_2)[X]$ where X denotes the attributes of r_1, is not valid anymore in the context of the certainty-based model. However, the semi-join can be defined in a sound way in this framework, see [13]. The key to the fact that join (and semi-join) can be easily handled in this model lies in the property that a tuple involving disjunctive values can produce at most one tuple in the result (due to the semantics of certainty).

Projection

Let r be a relation of schema (X,Y). The projection operation is straightforwardly defined as follows:

$$project(r, X) = \{\alpha/t.X \mid \alpha/t \in r \wedge \nexists \alpha'/t' \text{ s.t. } posbs(\alpha'/t'.X, \alpha/t.X)\}.$$

The only difference with respect to the definition of the projection in a classical database context concerns duplicate elimination, which is here based on the concept of "possibilistic subsumption" (using predicate *posbs*). Intuitively, an X-value of a tuple t is kept in the result if there is no other tuple t' with the same candidate values and a higher certainty level. More formally, letting $X = \{A_1, \ldots, A_n\}$, predicate *posbs* is defined as follows:

$$posbs(\alpha'/t'.X, \alpha/t.X) \equiv \forall i \in 1..n, scv(t.A_i) = scv(t'.A_i) \wedge cl(t.A_i) \leq cl(t'.A_i) \wedge \alpha \leq \alpha' \wedge ((\exists i \in 1..n, cl(t.A_i) < cl(t'.A_i)) \vee \alpha < \alpha').$$

The validity of the result before duplicate removal is guaranteed by the satisfaction of P2. As to the duplicate removal step, its soundness relies on the axioms of possibility theory. The definitions of the other relational algebraic operators in the certainty-based model can be found in [15].

4.3 Representation-Based Querying

The main motivation underlying the representation-based querying approach is to be able to exploit at a query level all the information available concerning the qualification of imperfectness in the data. In other words, one wants to be able to express conditions on the *descriptions* of ill-known data. Hereafter, we present a framework that was introduced in [5]. Representation-based queries can notably be used to:

- express conditions on specified sets of candidates (the specified set being a subset of a distribution representing an ill-known attribute value). The generic query is: "find the tuples such that all the elements of a specified subset of the candidate values (for a given attribute) satisfy a given condition",
- compute aggregates on the weighted sets corresponding to the representations of ill-known data (e.g., the cardinality of a specified subset of candidate values for a given attribute) and to use these aggregates inside conditions,

- compare a piece of data with a given vague pattern. In the representation-based querying framework, the comparison is based on the notion of synonymy of representations, contrary to the "value-based" framework where the comparison is founded on the notion of possibility/necessity of matching.

It is important to notice that representation-based queries are not just value-based queries expressed another way, but that they are queries of a different nature. A value-based criterion applying to an ill-known value has to be evaluated on each possible world associated with the attribute value (even though the explicit computation of those worlds is not always necessary, cf. Section 4.2.1), while a representation-based condition does not at all refer to worlds.

Example 7. We consider again a database containing aerial images of aircrafts (each image is supposed to represent a single aircraft), described by the set of attributes: (*#id*, *location*, *date*, *type*). The attributes *#id*, *location*, and *date* are supposed to take precise values whereas the attribute *type* describing the type of aircraft present in the picture will generally take imperfect values due to ambiguities in image interpretations. Examples of conditions involving one representation are:

- find the images which represent more likely a MiG29 than a MiG23,
- find the images such that all the candidates which are possible over 0.3 are of the type MiG,
- find the images for which at most 2 types of airplane are considered possible over 0.3,
- find the images for which the only best candidate is 'MiG29',
- find the images representing airplanes whose type is not precisely known (i.e., there are more than one candidate).$\diamond$

A language for representation-based conditions is described in [5]. In this framework, conditions involving two representations deserve a particular attention. Several methods have been proposed to compare possibility distributions or fuzzy sets and one can distinguish among two families of approaches. In the first family, a measure is used to evaluate the possibility degree of (approximate) equality between two imprecise values [17,18,24]. In the second family, what is measured is the extent to which two representations are globally close to each other [6,16,23,25]. In the representation-based querying framework, it is quite clear that only the second family of approach makes sense. Let us consider an attribute A and two items x and y whose A-values are ill-known. Let us denote by $\pi_{A(x)}$ and $\pi_{A(y)}$ the possibility distributions to be compared. Let D be the domain of attribute A. First, let us recall the expression of strict equality:

$$\forall d \in D,\ \pi_{A(x)}(d) = \pi_{A(y)}(d).$$

Several authors have proposed to relax the preceding measure into a measure of approximate equality. Raju and Majumdar [25] define the fuzzy equality measure, denoted EQ, in the following way:

$$\mu_{EQ}(\pi_{A(x)}, \pi_{A(y)}) = min_{u \in D}\ \psi(\pi_{A(x)}(u),\ \pi_{A(y)}(u))$$

where ψ is a resemblance relation (i.e., reflexive and symmetric) over [0, 1]. An alternative approach consists in defining the similarity of two fuzzy sets (two possibility distributions in our case) A and B as a function of $A \cap B$, $B - A$ and $A - B$. This approach is studied in particular by Bouchon-Meunier *et al.* [16] where different kinds of measures of comparison are considered.

Example 8. Let us consider the following description of an image I_1:

$$I_1 = \langle 27, Krasnoyarsk, \text{1992-12-01}, \{1/Su27, 1/Su30, 0.7/Mig29, 0.2/Yak130\}\rangle.$$

Let us consider the query: "find the pictures taken over Krasnoyarsk in 1992 representing an airplane similar to the one in image I_1" and let us assume that the database contains notably the following description:

$$I_2 = \langle 51, Krasnoyarsk, \text{1992-04-15}, \{1/Su30, 0.9/Mig29, 0.8/Su27, 0.4/Mig23\}\rangle.$$

If strict equality were used, it is clear that image I_2 would not belong to the result. Using Raju-Majumdar's measure of approximate equality with $(a, b) = 1 - |a - b|$, the matching degree between I_1 and I_2 is equal to:

$$EQ(I_1.type, I_2.type) = min(0.8\,(Su27), 1\,(Su30), 0.8\,(Mig29), 0.8\,(Yak130), 0.6\,(Mig23)) = 0.6.\diamond$$

On the other hand, these measures can be used to compare an ill-known attribute value D with a linguistic label P. The basic idea is the same: one evaluates the extent to which the value and the linguistic label represent the same concept. For example, let us consider a possibility distribution D representing John's age and a linguistic label P = "middle-aged" (represented by a fuzzy set). While the value-based querying approach aims at assessing the extent to which John is possibly (resp. necessary) middle-aged, the representation-based approach can be used to measure the extent to which the description of John's age and the linguistic label "middle-aged" are close to each other. This approach is especially useful in the context of applications where user queries can be conveniently expressed by means of linguistic terms defined on continuous domains. Lastly, the concept of representation-based comparison can be used to define the notions of representation-based intersection, union and difference in a straightforward manner.

5 Conclusion

In this paper, we have reviewed different types of queries that can be addressed to a database containing imprecise values represented in the possibilistic framework. We have distinguished three main lines: i) the initial approach proposed by Prade and Testemale which is intended for building "events" and their associated possibility and necessity degrees from data, ii) works based on the possible worlds semantics with two data models: the full possibilistic model where queries are constrained and the certainty-based model which offers the richness of the entire relational algebra, iii) queries where the conditions bear on the representation of imprecise data. The focus has been put

on the semantic aspects, not on implementation and performances due to space limits. However, most of the operators proposed are very similar to those defined in regular database systems and reasonable performances can be expected. Among others, future works could concern queries involving preferences in the spirit of [8,14].

References

1. Abiteboul, S., Hull, R., Vianu, V.: Foundations of Databases. Addison-Wesley, Reading (1995)
2. Abiteboul, S., Kanellakis, P., Grahne, G.: On the representation and querying of sets of possible worlds. Theoretical Computer Science 78, 159–187 (1991)
3. Antova, L., Jansen, T., Koch, C., Olteanu, D.: Fast and simple processing of uncertain data. In: Proc. of 24th Int. Conf. on Data Engineering (ICDE 2008), pp. 983–992 (2008)
4. Benjelloun, O., Das Sarma, A., Halevy, A., Widom, J.: ULDBs: Databases with uncertainty and lineage. In: Proc. of VLDB 2006, pp. 953–964 (2006)
5. Bosc, P., Duval, L., Pivert, O.: Value-based and representation-based querying of possibilistic databases. In: Bordogna, G., Pasi, G. (eds.) Recent Research Issues on Fuzzy Databases, pp. 3–27. Physica-Verlag, Heidelberg (2000)
6. Bosc, P., Pivert, O.: On the comparison of imprecise values in fuzzy databases. In: Proc. of the 6th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 1997), Barcelona, Spain, pp. 707–712 (1997)
7. Bosc, P., Pivert, O.: About the projection operator in a possibilistic database framework. In: Proc. of the 21st Int. Conf. of the North American Fuzzy Information Processing Society (NAFIPS 2002), New Orleans, Louisiana, USA, pp. 371–376 (2002)
8. Bosc, P., Pivert, O.: From boolean to fuzzy algebraic queries in a possibilistic database framework. In: Proc. of the 13th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2004), Budapest, Hungary (2004)
9. Bosc, P., Pivert, O.: On a strong representation system for imprecise relational databases. In: Proc. of the 10th International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU 2004), Perugia, Italy, pp. 1759–1766 (2004)
10. Bosc, P., Pivert, O.: About projection-selection-join queries addressed to possibilistic relational databases. IEEE Trans. on Fuzzy Systems 13(1), 124–139 (2005)
11. Bosc, P., Pivert, O.: Vacuity-oriented generalized yes/no queries addressed to possibilistic databases. In: Bordogna, G., Psaila, G. (eds.) Flexible Databases Supporting Imprecision and Uncertainty, pp. 55–74. Springer (2006)
12. Bosc, P., Pivert, O.: About yes/no queries against possibilistic databases. International Journal of Intelligent Systems 22, 691–722 (2007)
13. Bosc, P., Pivert, O., Prade, H.: A Model Based on Possibilistic Certainty Levels for Incomplete Databases. In: Godo, L., Pugliese, A. (eds.) SUM 2009. LNCS, vol. 5785, pp. 80–94. Springer, Heidelberg (2009)
14. Bosc, P., Pivert, O., Prade, H.: A possibilistic logic view of preference queries to an uncertain database. In: Proc. of the 19th IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2010), Barcelona, Spain, pp. 379–384 (2010)
15. Bosc, P., Pivert, O., Prade, H.: An uncertain database model and a query algebra based on possibilistic certainty. In: Proc. of the 2nd International Conference on Soft Computing and Pattern Recognition (SoCPaR 2010), Cergy-Pontoise, France (2010)
16. Bouchon-Meunier, B., Rifqi, M., Bothorel, S.: Towards general measures of comparison of objects. Fuzzy Sets and Systems 84, 143–153 (1996)

17. Chen, G., Kerre, E., Vandenbulcke, J.: A general treatment of data redundancy in a fuzzy relational data model. Journal of the American Society for Information Science 43, 304–311 (1992)
18. Cubero, J., Vila, M.: A new definition of fuzzy functional dependency in fuzzy relational databases. International Journal of Intelligent Systems 9, 441–448 (1994)
19. Dalvi, N., Suciu, D.: Management of probabilistic data: Foundations and challenges. In: Proc. of PODS 2007, pp. 1–12 (2007)
20. Dubois, D., Prade, H.: Necessity measures and the resolution principle. IEEE Trans. Syst., Man and Cyber. 17(3), 474–478 (1987)
21. Dubois, D., Prade, H.: Possibility Theory. Plenum, New York (1988)
22. Imielinski, T., Lipski, W.: Incomplete information in relational databases. J. of the ACM 31(4), 761–791 (1984)
23. Liu, W.: The fuzzy functional dependency on the basis of the semantic distance. Fuzzy Sets and Systems 59, 173–179 (1993)
24. Prade, H., Testemale, C.: Generalizing database relational algebra for the treatment of incompleteuncertain information and vague queries. Information Sciences 34, 115–143 (1984)
25. Raju, K., Majumdar, A.: Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. ACM Transactions on Database Systems 13, 129–166 (1988)
26. Zadeh, L.: Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems 1(1), 3–28 (1978)