About Various Uses of Algebraic Queries Against Possibilistic Databases

Patrick Bosc, Olivier Pivert
IRISA/ENSSAT
Technopole Anticipa BP 447
22305 Lannion Cedex France
e-mail: {bosc, pivert}@enssat.fr

Abstract—**This paper is concerned with the handling of imprecise information in possibilistic relational databases. Since the processing of general algebraic queries raises problems in terms of tractability in the presence of imprecise information, restricted algebraic queries can only be considered. It turns out that the result obtained is a compact representation of all the more or less possible results in terms of regular databases. In this paper, a series of queries which are more user-oriented and can be seen as built on top of restricted algebraic queries are pointed out.**

I. Introduction

In this paper, relational databases are extended to the case where some attribute values are imprecisely known. Imprecise information can appear in diverse situations such as data warehouses, forecasts, incomplete archives, or systems where information issued from automated recognition is stored. Different formalisms can be used to represent imprecise information (see for instance [3, 5]), and the possibilistic setting [8] is assumed in the rest of the paper.

A key question is to define a sound semantics for queries addressed to imprecise databases. Since imprecise data are represented as (possibly infinite) sets of acceptable candidates, an imprecise database can be seen as a set of regular databases, called worlds, associated with a choice for each attribute value. This approach provides a rational starting point for the definition of a query in the sense that its result is a compact representation of the results obtained in each world. Unfortunately, this approach is intractable due to the huge (possibly infinite) number of worlds. This observation leads to consider only specific queries which can be processed directly against the possibilistic database, while delivering a result equivalent to the one defined in terms of worlds.

A compact calculus valid for a subset of the relational algebra has been devised (see [1, 2] for details). Interestingly, it applies to queries containing some binary operations allowing for the composition of relations. In this context, the result of a query is a possibilistic relation whose interpretations correspond to more or less possible results, equivalent to those which would have been obtained with a calculus applied to worlds. This achievement is interesting from a methodological point of view, but the use of this type of result by a final user can be somewhat delicate. So, it becomes convenient to define queries which are more specialized to fit user needs. Different queries are presented which are intended to meet this goal.

The structure of the paper is the following. In section II, the notion of possibilistic relational databases is introduced. Then, the data model requested for a valid compact processing of algebraic queries is described in section III. It is based on two simple appropriate extensions of the basic model advocated by Prade and Testemale [6]. Section IV is devoted to a brief presentation of the version of algebraic operators which can be processed in a compact way. In section V, different types of queries making use of algebraic queries are introduced. Finally, the conclusion summarizes the contributions of the paper and draws some lines for future works.

II. Possibilistic Relational Databases and Worlds

A. Possibility theory : some key notions

Possibility theory [8] provides an ordinal model for uncertainty where imprecision is represented by means of a preference relation coded by a total order over the possible situations. This framework is closely related to fuzzy set theory [7] since the idea is to constrain the values taken by a variable thanks to a (normalized) fuzzy set called a possibility distribution. More formally, a possibility distribution is an application π from a domain X to the unit interval [0, 1] and $\pi(a)$ expresses the degree to which a is a possible value for the considered variable. The normalization condition imposes that at least one of the values of the domain (a_0) is completely possible, i.e.:

$$\pi(a_0) = 1.$$

This setting is particularly suited to take into account uncertainty represented by linguistic terms such as "high", "large", "expensive" and so on. When the domain is discrete, a possibility distribution can be written:

$$\{\pi_1/a_1 + \ldots + \pi_n/a_n\}$$

where a_i is a candidate value and π_i its possibility degree.

In the rest of this paper, only finite possibility distributions are taken into account. Any event E defined on the powerset of X is characterized by two measures Π and N. The axioms related to the measure of possibility Π are the following:

$$\Pi(X) = 1 \text{ (which requires the normalization condition)},$$

$\Pi(\emptyset) = 0$,

$$\Pi(E1 \cup E2) = \max(\Pi(E1), \Pi(E2)) \quad (1)$$

and the measure of possibility of the event E is derived from the possibility distribution associated with the concerned variable in the following way:

$$\Pi(E) = \max_{x \in E} \pi(x).$$

The possibility of the conjunction of two non interactive events is given by [4]:

$$\Pi(E1 \cap E2) = \min(\Pi(E1), \Pi(E2)) \quad (2).$$

The only relationship between the possibility of **not** E (the opposite event of E) and that of E is:

$$\max(\Pi(E), \Pi(\textbf{not } E)) = 1$$

which entails that if $\Pi(E) = 1$, nothing can be said for $\Pi(\textbf{not } E)$, which can range from 0 to 1. In order to have a better characterization of the event E, the measure of certainty (also called measure of necessity) N has also been introduced inside the possibilistic framework:

$$N(E) = 1 - \Pi(\textbf{not } E).$$

In other words, the less possible **not** E, the more certain E. Due to the duality between these two measures, the following formula holds in the general case:

$$N(E1 \cap E2) = \min(N(E1), N(E2)) \quad (3).$$

In addition, if E1 and E2 are two non interactive events:

$$N(E1 \cup E2) = \max(N(E1), N(E2)) \quad (4).$$

As far as regular (i.e., non fuzzy) events are concerned, it can be proven that:

$$\Pi(E) < 1 \Rightarrow N(E) = 0 \quad (5).$$

Then, these two measures provide a total order over the set of events which can be ordered according to Π for those which are not at all certain and according to N for those which are completely possible.

B. Possibilistic databases

In contrast to a regular database, a possibilistic relational database **D** may have some attributes which take imprecise values. In such a case, a possibility distribution is used to represent all the more or less acceptable candidates for the attribute.

The first version of a possibilistic database model was introduced by Prade in the mid 80s [6]. From a semantic point of view, a possibilistic database **D** can be interpreted as a set of usual databases (also called worlds), denoted by rep(**D**), each of which being more or less possible (one of them is supposed to correspond to the actual state of the universe modeled). This view establishes a semantic connection between possibilistic and regular databases. It is particularly interesting since it offers a canonical approach to the definition of queries addressed to possibilistic databases as will be seen later (section V). Any world W_i is obtained by choosing a candidate value in each possibility distribution appearing in **D** and its degree of possibility is the minimum of those of the candidates taken (thanks to formula 2).

im	#i	ap	date	place
	i1	a1	{1/d1 + 0.7/d3}	c1
	i3	{1/a3 + 0.3/a4}	d1	c2

Example 1. Let us consider the possibilistic database **D** involving two relations: im and pl whose respective schemas are IM(#i, ap, date, place) and PL(ap, lg, sp). Relation im describes satellite images of airplanes and each image, identified by a number (#i), taken on a certain location (place) a given day (date) is supposed to include a *single* (possibly ill known) airplane (ap). Relation pl gives the length (lg) and maximal speed (msp) of each airplane and is a regular (precise) relation.

With the extension of im given before, four worlds can be drawn, since there are two candidates for date (resp. ap) in the first (resp. second) tuple of im. Each of these worlds involves the relation pl which has only precise values and one of the four regular relations issued from the possibilistic relation im above.♦

III. An Extended Possibilistic Data Model

A. Objective

As mentioned before, a calculus based on the processing of the query (Q) against worlds is intractable and a *compact* approach to the calculus of the answer to Q must be found out. It is then necessary to be provided with both a data model and operations which have good properties: i) the data model must be *closed* for the considered operations, and ii) any query (applying to the possibilistic database **D**) must be processed in a *compact* way. In addition, its result must be a compact representation of the results of this query if it were applied to all the interpretations (worlds) drawn from **D**, i.e.:

$$\text{rep}(\text{Qc}(\mathbf{D})) = \text{Q}(\text{rep}(\mathbf{D})),$$

where rep(**D**) denotes the set of worlds associated with **D** and Qc stands for the query obtained by replacing the operators of Q by their compact versions. This property characterizes data models called *strong representation systems*.

It turns out that the initial relational possibilistic model cannot comply with this property in at least two respects (notably for the selection): i) the recovery of "missing tuples", and ii) the accounting for dependencies between candidate values.

B. Representing possibly missing tuples

There is a need at the compact level for expressing that some tuples can have no representative in some worlds. Indeed, some operations (e.g., selections) lead to discard candidate values from a distribution, but one must be able to compute the degree of any different world of the answer, including those for which no representative of some tuples are taken (because they correspond to candidates that have been discarded).

A simple solution is to introduce a new attribute, denoted by **N** (valued in [0, 1]), which states whether or not it is legal to build worlds where no representative of the corresponding tuple is present, and, if so, the influence of this choice in terms of degree of possibility. The value of **N** associated with a tuple t expresses the *certainty of the presence* of a representative of t in any world. A tuple is denoted by a pair N/t where N equals 1 for tuples of initial possibilistic relations as well as when no alternative has been discarded.

Exemple 2. Let us consider the following extension of the possibilistic relation im:

im	#i	ap	date	place
	i1	B-727	d1	c1
	i2	ATR-72	d1	c2
	i3	{1/B-727 + 0.7/ATR-42}	d2	c4
	i4	{1/B-727 + 1/B-747 + 0.4/ATR-72}	d2	c2

The selection based on the condition "ap = B-727" leads to discard the candidates which are different from this desired value. Thanks to the introduction of attribute **N**, the result of the selection is:

res	#i	ap	date	place	N
	i1	B-727	d1	c1	1
	i3	B-727	d2	c4	0.3
	i4	B-727	d2	c2	0

From this result, it is possible to derive the interpretation made of the single tuple <i1, B-727, d1, c1> whose degree of possibility is: min(1, 1 – 0.3, 1 – 0) = 0.7. Let us notice that this result is the one obtained from the interpretation im1 of im where ATR-42 (resp. B-747) is chosen in the third (resp. fourth) tuple. The degree of possibility attached to im1 is: min(1, 1, 0.7, 1) = 0.7, which equals the value obtained with the compact calculus.♦

C. Multiple attribute possibility distributions

Another aspect of the data model is related to the fact that it is sometimes necessary to express relationships (dependencies) between candidate values coming from different attributes in a same tuple.

For instance, let us consider a given tuple t where the two attributes A and B take the imprecise values t.A = $\{a_1, a_2\}$ and t.B = $\{b_1, b_2, b_3\}$. If an operation retains only the pairs (a_1, b_1) and (a_2, b_3), it is impossible to represent this situation with a Cartesian product of subsets of t.A on the one hand and t.B on the other hand, and the correct associations must be explicitly represented. This requires that the model incorporates attribute values defined as possibility distributions over *several domains*. This is feasible in the relational framework thanks to the concept of a *nested relation*.

So doing, exclusive candidates are represented as weighted tuples. Therefore, level-one relations keep their *conjunctive* meaning, whereas nested relations have a *disjunctive* interpretation.

Example 3. Let us consider the relation r whose schema is R(A, B, C, D), containing the tuple: t = 1/<a1, {1/8 + 1/10 + 0.7/12}, {1/9 + 0.4/15}, d1>. If the selection B < C is performed on r, the resulting relation must include a nested relation X(B, C) over attributes B and C and the tuple obtained from t is t' = 0/<a1, {1/<8, 9> + 0.4/<8, 15> + 0.4/<10, 15> + 0.4/<12, 15>}, d1>. The value of attribute **N** is 0 because the completely possible pair (10, 9) has been discarded.♦

Finally, the extended possibilistic relational model incorporates two new features: an extra attribute **N** and the possibility distributions defined over several attributes. An example of such a relation is illustrated hereafter.

Example 4. Let us consider the following intermediate relation int-r:

int-r	#i	ap	X date place	N
	i1	B-727	{1/<d1, c1> + 0.7/<d1, c2> + 0.4/<d3, c2>}	1
	i3	B-727	<d1, c2>	0.3
	i4	{0.4/B-737}	{0.3/<d3, c2>}	0

This relation is associated with 12 worlds since the first tuple admits 3 interpretations, the second and third ones have two interpretations among which one where they have no representative.♦

IV. Compact Version of the Algebraic Operators

A. Introduction

In order to meet the objective of a compact processing of algebraic queries, the operators must be adapted so as to accept compact relations (as defined in the previous section) both as inputs and outputs. It turns out that only operations such that an input tuple participates in the production of at most one element of the result, can be expected to admit a compact version (see [1, 2]).

As a consequence, the intersection, the difference and the Cartesian product (then the join in the general case) are discarded and the four acceptable operators are now dealt with. Due to space limitations, there is no room for a technical presentation of these operators. We limit ourselves to a brief introduction and their functioning is illustrated by a worked out example in subsection IV.C.

B. Unary operations

The three roles of the selection are:

1) the removal of unsatisfactory candidate values,

2) the computation of the degree of certainty attached to each output tuple,

3) the introduction of appropriate nested relations in the output relation if needed.

The role of the projection in the regular case is to remove undesired attributes. In the context of the extended relational model presented in section III, four aspects have to be taken into account for the projection of a relation r on a subset of its attributes in the presence of imprecise values:

1) the impact of the projection on the value of the attribute **N** in each tuple. Indeed, this value is not affected by a projection and it remains unchanged in the output tuple,

2) the role of duplicates. Duplicates must be kept in level-one relations in order to recover *all the legal interpretations* of the projected relation. For instance, if relation r whose schema is R(A, B) contains the tuples $<\{\pi_1/a_1 + \ldots + \pi_n/a_n\}, b_1>$ and $<\{\pi_1/a_1 + \ldots + \pi_n/a_n\}, b_2>$, the projection over A results in a relation with two identical tuples. This makes it possible to have a world with two tuples $<a_i>$ and $<a_j>$, which would be quite impossible with a single tuple kept in the projection. On the contrary, tuples in a nested relation have a disjunctive meaning since they represent alternative candidates. Therefore, duplicates are meaningless in nested relations and according to possibility theory (formula 1), the highest possibility degree is retained if several tuples become the same after projection,

3) the structure of the resulting relation. If all the attributes of a nested relation are suppressed, the nested relation totally disappears. If only one of the attributes of a nested relation is retained, the nested relation becomes a level-one attribute, otherwise the nested relation is projected as usually done,

4) the possibility degrees: when an attribute with an imprecise value or a nested relation disappears, the possibility of the most possible candidate is aggregated via min with that of one of the remaining attributes. So doing, it is guaranteed that no world can be drawn with a possibility greater than that of the corresponding world before projection.

C. Binary operations

Beyond selections and projections, two binary operations can be processed in a compact fashion. Although there is no hope for defining a compact version of the join, it turns out that a specific join, namely the fk-join, is acceptable. The fk-join allows for the composition of a *possibilistic* relation r of schema R(W, Z), where W and Z may take imprecise values, and a *regular* relation s whose schema is S(W, Y) where the functional dependency W → Y holds. So, W is the key of s and then a *foreign key* of r. The fk-join consists in completing tuples of r by adding the image of the W-component (i.e., the associated value of attribute Y via W → Y).

By definition, this leads to a resulting relation involving the nested relation X(W, Y), which "connects" the pairs of candidates over W and Y. The degree of possibility of any structured candidate value is the one of the value that has been completed. Similarly to the selection, **N** is updated to keep track of the most possible candidate value for which no match occurred.

Last, the union of two relations whose schemas are compatible (i.e., have attributes pairwise compatible even if nested relations are not the same) keeps all the tuples issued from the two input relations without any duplicate removal. This mechanism ensures that all (and only) the correct interpretations can be recovered, provided that the two input relations are *independent* (do not come from a same relation).

Example 5. This example is intended for illustrating the overall functioning of the procedure designed for answering algebraic queries addressed to possibilistic databases. Let us consider the possibilistic database composed of the relations im1(IM), im2(IM) and pl(PL) where IM and PL are the schemas introduced in example 1. The two relations im1 and im2 are assumed to contain images of airplanes taken by two *distinct* sources (e.g., satellites). Let us take the algebraic query Q:

fk-join(union(select(im1, date ∉ {d3, d4}),
select(im2, date ∉ {d3, d4}),
select(pl, msp > 900)),
{ap}, {ap}).

With the extensions given hereafter:

pl

ap	lg	msp
a1	20	1000
a2	25	800
a3	18	600
a4	20	1200
a5	20	1000

im1

#i	ap	date	place	N
i1	a3	{1/d1 + 0.7/d3}	c1	1
i2	{1/a2 + 0.7/a1}	d1	c2	1

im2

#i	ap	date	place	N
i3	{1/a4 + 1/a5}	{0.6/d4 + 1/d1}	c3	1

the selections on im1, im2 and pl create the following intermediate relations:

pl'

ap	**lg**	**msp**
a1	20	1000
a4	20	1200
a5	20	1000

im'1

#i	**ap**	**date**	**place**	**N**
i1	a3	{1/d1}	c1	0.3
i2	{1/a2 + 0.7/a1}	d1	c2	1

im'2

#i	**ap**	**date**	**place**	**N**
i3	{1/a4 + 1/a5}	{1/d1}	c3	0.4

After the union of im'1 and im'2, the fk-join introduces a nested relation over attributes ap, lg and msp, and the result delivered is:

res

#i	**X**	**date**	**place**	**N**
	ap lg msp			
i2	{0.7/<a1, 20, 1000>}	d1	c2	0
i3	{1/<a4, 20, 1200> + 1/<a5, 20, 1000>}	{1/d1}	c3	0.4

♦

V. Requests Based on Algebraic Queries

A. Introducing a post processing

On the basis of the previous version of the algebraic operators, a query can be processed in a tractable way since operations are performed in a compact fashion. However, one may wonder about the usability of the result delivered by such a query, i.e., of a compact relation as such. We think that a convenient direction is to provide users with queries which are close to their needs and then to call on (embedded) algebraic queries. Hereafter, some prototypical types of such user-oriented queries are highlighted. Their evaluation is based on a two-step mechanism:

1) a *compact* calculus of the associated algebraic query, which builds a compact relation according to the procedure depicted in the preceding section,

2) a post processing producing the final answer (i.e., the answer to the user's query). This latter phase depends strongly on the type of user query under consideration and its complexity will vary accordingly.

B. Extended yes/no queries

Type 1 extended yes/no queries have the general format:

"to what extent is it *possible* and *certain* that t belongs to the answer to Q?"

where t is a given target tuple. They are intended for generalizing queries of the form: "is it true that t belongs to the answer to Q?", applicable in the presence of precise data, whose answer is either yes, or no. Their processing consists in:

1) evaluating the algebraic query Q,

2) determining the possibility of the most possible world of the answer where t is involved and that of the most possible world where t is not present. This post processing requires a single scan over the result produced by step 1.

Type 2 extended yes/no queries are statements of the form:

"to what extent is it *possible* and *certain* that the answer to Q is not empty?".

which come from regular queries of the form: "is it true that the answer to Q is not empty?". Their processing is very similar to that of type 1 yes/no queries.

For instance, let us consider the query:

"to what extent is it *possible* and *certain* that there exists at least one shot with an aircraft of maximal speed greater than 900 km/h, taken on a date different from d3 and d4?"

addressed to the possibilistic database of example 5. More formally this query is:

"to what extent is it *possible* and *certain* that the result of Q is non empty?",

where Q is the query dealt with in example 5. From the table res obtained, it is possible to derive a world with the tuple <i3, a4, 20, 1200, d1, c3> which is completely possible. Then, the possibility that the answer to Q is not empty equals 1. Similarly, an empty world can be drawn from res with the maximal possibility: min(1 – 0, 1 – 0.4) = 0.6, and the certainty that the answer to Q is not empty is: 1 – 0.6 = 0.4.

C. Multiple tuple possibilistic queries

Multiple tuple possibilistic queries of the form:

"to what extent is it *possible* (and *certain*) that tuples $\{t_1, \ldots, t_n\}$ jointly belong to the answer to Q ?"

are also of interest. From the relation produced by the compact processing of query Q, one has to identify the most possible world (if any) which involves the set $\{t_1, \ldots, t_n\}$. This requires a "try and error" (or a "branch and bound") technique, which is more complex than the previously mentioned post processings, but whose complexity can be expected to remain reasonable in practice.

D. Cardinality-based queries

Cardinality-based queries of the form:

"to what extent is it *possible* (and *certain*) that the answer to Q has at least (at most, . . .) n elements?"

may be seen as a variation of the previous type of query. The post processing of the compact result of Q aims at the determination of the possibility of worlds involving a certain number of elements.

The problem is to deal with tuples of the resulting relation which can give birth to representatives which are indeed duplicates. For instance, if the result of the compact processing of the algebraic query Q is:

res

A	B	N
{1/a1 + 0.6/a2}	b	0.3
a1	b	1

the degree of possibility that the answer contains at least 2 tuples cannot be obtained by taking the most possible representative of the two tuples of res because they are identical (<a1, b>).

That is the reason why, here again, the procedure attached to the post processing must rely on a "try and error" (or a "branch and bound") technique in order to identify the most satisfactory worlds with respect to the desired cardinality.

For instance, the query:

"to what extent is it *possible* that there exists *at least two* shots with an aircraft of maximal speed greater than 900 km/h, taken on a date different from d3 and d4?"

reduces to determining the possibility of a world issued from table res delivered in example 5, which contains at least 2 tuples. It turns out that there is such a world (indeed involving exactly two tuples), which is possible at the degree:

$$\min(0.7, 1) = 0.7.$$

If "at least two" is replaced by "at least 3" (or more) in the query, the degree obtained would be 0.

In the same spirit, it is possible to imagine queries where "count" is replaced by another aggregate such as average, maximum, etc. The following query illustrates the use of the aggregate "minimum":

"to what extent is it possible and certain that the smallest maximal speed of aircrafts appearing on shots taken before date d1 is over 950 km/h?".

The treatment of such a query is quite similar to that suggested before for cardinality-based queries. Once the algebraic query is processed, its result is exploited in order to find out the most possible representative where the desired condition is satisfied (resp. not satisfied for determining the necessity).

VI. Conclusion

This paper addresses the issue of querying relational databases where some attribute values are imprecise and represented as possibility distributions. For tractability reasons, *restricted* algebraic queries have been investigated.

Their main characteristic is to be processed in a compact fashion and then to lead to acceptable performances. The price to pay is in terms of authorized operators, namely only the selection, projection, union and the fk-join (a specific join). The result of such a query is a possibilistic table, which is equivalent to the set of resulting relations which would be obtained if the query were processed against each interpretation of the possibilistic database.

However, we guess that the form of such a result is not very convenient for a final user. This is why a variety of queries, expected to better fit user needs, is introduced. Their treatment is based on two steps, the first one being the evaluation of a restricted algebraic query against the possibilistic database.

This work opens different lines for future research. One of them is related to the performances obtained for the family of queries that have been considered, especially taking into account the specificity of the post processing in each case. Another direction of research concerns the influence of the model of uncertainty chosen to represent ill-known information. In other words, what will happen if the possibilistic database is replaced by a probabilistic one? How can the queries envisaged before be reformulated in the probabilistic context and can they still be processed in a similar compact way? Another subject of interest concerns the introduction of *fuzzy* restricted algebraic queries instead of Boolean ones.

References

[1] P. Bosc, O. Pivert, Towards an algebraic query language for possibilistic databases, *12th Conference on Fuzzy Systems (FUZZ-IEEE'03)*, pp. 671-676, 2003.

[2] P. Bosc, O. Pivert, About projection-selection-join queries addressed to possibilistic relational databases, *IEEE Transactions on Fuzzy Systems*, to appear.

[3] P. Bosc, H. Prade, An introduction to the treatment of flexible queries and uncertain or imprecise databases, in: *Uncertainty Management in Information Systems*, A. Motro and P. Smets (Eds), Kluwer Academic Publishers, pp. 285-324, 1997.

[4] D. Dubois, H. Prade, Fuzzy Sets and Systems – Theory and Applications. Academic Press, 1980.

[5] T. Imielinski, W. Lipski, Incomplete information in relational databases, *Journal of the ACM*, vol. 31, pp. 761-791, 1984.

[6] H. Prade, C. Testemale, Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. *Information Sciences*, vol. 34, pp. 115-143, 1984.

[7] L.A. Zadeh, Fuzzy sets, *Information and Control*, vol. 8, pp. 338-353, 1965.

[8] L.A. Zadeh, Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, vol. 1, pp. 3-28, 1978.