



An introduction to the fuzzy set and possibility theory-based treatment of soft queries and uncertain of imprecise databases

Patrick Bosc, Henri Prade

► To cite this version:

Patrick Bosc, Henri Prade. An introduction to the fuzzy set and possibility theory-based treatment of soft queries and uncertain of imprecise databases. Amihai Motro; Philippe Smets. Uncertainty Management in Information Systems: From Needs to Solutions, Springer US, pp.285–324, 1997, 978-1-4613-7865-5. 10.1007/978-1-4615-6245-0_10 . hal-04034581

HAL Id: hal-04034581

<https://hal.science/hal-04034581>

Submitted on 17 Mar 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An introduction to the fuzzy set and possibility theory-based treatment of soft queries and uncertain or imprecise databases

Patrick BOSC

I.R.I.S.A/E.N.S.S.A.T.
Université de Rennes I
6 rue de Kérampont – BP 447
22305 Lannion Cedex – France

Henri PRADE

I.R.I.T.
Université Paul Sabatier
118 route de Narbonne
31062 Toulouse Cedex – France

Abstract

In this paper, it is shown that fuzzy sets and possibility theory provide an homogeneous framework for the representation of both imprecise/uncertain information and soft queries with a flexible interpretation. Incompletely known information as well as flexible query handling capabilities are expected to extend the range of applications for future database management systems. The term fuzzy databases which is extensively used in the specialized literature covers several different meanings which are reviewed. A special emphasis is put on flexible queries addressed to regular databases. Such queries enables the user to easily express preferences among more or less admissible attribute values. Several approaches for introducing flexibility, including fuzzy sets, are compared. A query language based on SQL is outlined and some issues related to query processing are discussed. In addition, possibility theory proves to be useful for representing imperfectly known data and soft constraints. Pattern matching is extended for handling this kind of data, as well as flexible queries.

1. Introduction

The information to be stored in a data base is not always precise and certain. Information about attribute values may be even missing for objects of interest (this corresponds to the null value "unknown"). Other types of null values (e.g., Biskup, 1980) are also encountered when an attribute does not apply to a particular object or when we do not know if the value is missing or does not exist. When the only available information is imprecise, uncertain or even vague it seems rather natural to try to represent it, as it is, and to store it in a data base in order to use it to answer queries of interest as far as possible.

Another data base issue which may be also connected with the idea of uncertainty, is the handling of unsharp queries, of queries which are not clear-cut. There may be several reasons for such queries. First the way the query is stated may be ambiguous because for instance it is expressed in natural language and we do not know to what, some part of the query refers. Some contextual and pragmatic knowledge may then help to choose the most plausible interpretation. Natural language queries may also involve words or phrases with vague meanings. The user may also be unsure about what he is looking for. In all these cases the unsharpness can be viewed as a defect of the query. However the presence of vague predicates in a query may be also motivated by the need of expressing gradual preferences between admissible values. Then the query is made flexible for expressing tolerance.

Fuzzy set and possibility theory (Zadeh, 1965, 1978) offer a unified framework which enables us to provide a solution both to the handling of flexible queries and to the management of imprecise and uncertain information. A fuzzy set F is an extension of the idea of a regular set viewed

in terms of membership function. It describes a subpart of a universe U whose boundaries are not strictly defined and a grade of membership (valued in $[0,1]$) is attached to each element of U . In so doing, there is a gradual transition between full membership and exclusion and it is then possible to have a better representation of gradual properties, vague classes, approximate descriptions currently used in natural languages when speaking of the real world. As a consequence, if we consider a precise value x in U , it is possible to estimate the extent to which x is compatible with the concept represented by F . In other situations, the issue is rather to represent an ill-known value x and, in this case, the notion of possibility distribution is used to express a restriction on the more or less possible values of x and to specify the extent to which a given element of U is possible as being the actual value of x . Clearly, fuzzy sets and possibility distributions encode preferences on a universe, but it should be emphasized that they deal with two different issues as illustrated by Figure 1. Indeed, depending on the considered situation, Figure 1 may either mean that the value "a" (e.g., John's salary) is compatible at degree .8 with the concept represented by the membership function of a fuzzy set ("high" for instance) if John's salary is precisely known, or it may mean that it is possible at degree .8 that "a" is the actual value of an ill-known piece of data whose possible values are described by the possibility distribution in bold in Figure 1 (e.g., we only know that John's salary is high). Possibility distributions restrict the possible values of variables on universes of mutually exclusive values, while fuzzy sets represent gradual properties whose satisfaction may be a matter of degree.

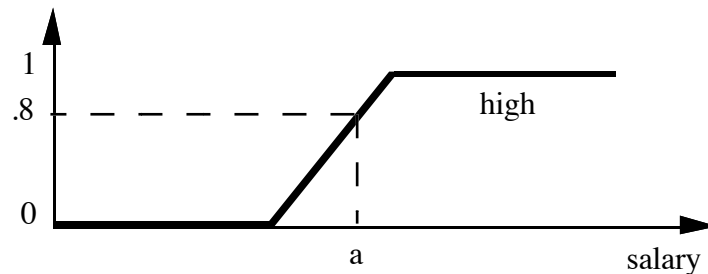


Figure 1 : Fuzzy set or possibility distribution ?

The paper gives a general introduction to fuzzy set and possibility theory-based approaches, provides an overview of the available methods, and refers to the existing literature where more details can be found. The paper is organized in the following way. Section 2 defines the basic vocabulary and introduces the fundamental concepts for characterizing the various forms of imperfect information. Section 3 examines what are the main different kinds of fuzzy data bases. Section 3 also briefly considers the corresponding issues in documentary information retrieval systems. Section 4 is devoted to the treatment of flexible queries. Section 5 deals with the management of imprecise and uncertain pieces of information in a data base. Section 6 discusses integrity constraints and especially functional dependencies whose expressions involve tolerance, graduality or uncertainty. In the concluding remarks, other topics related to uncertainty and imprecision issues in data base systems are briefly mentioned, in particular, the updating of the content of a data base when new information is provided or time is passing, the management of fuzziness and uncertainty in object-oriented representations, and the summarization of data.

2. Imperfect Information : Vocabulary

The purpose of this section is to define the main terms which are in use when referring to imperfect information. By information, we mean data (facts) as well as general knowledge (rules). We successively distinguish four main types of defects which may pervade information, namely : uncertainty, imprecision, vagueness and inconsistency. The distinction between imprecision and uncertainty has been particularly advocated by Dubois and Prade (1988a).

2.1. Uncertainty

Uncertainty refers to the lack of available information about the state of the world for determining if a classical statement (which can only be true or false) is actually true or false. Examples of such statements are : "It will rain at 5 p.m. to-day", "Tweety flies" (knowing only that Tweety is a bird and that birds usually fly), "The number of inhabitants in Palma de Mallorca is over 500.000" (but the person who gave me this information is not fully reliable).

In such situations, the best that we can do is to try to estimate the tendency of the statement to be true (or to be false) and several frameworks are possible : i) numerical approaches such as probability theory, possibility theory, belief functions, and more ad hoc certainty factor-based techniques, ii) purely symbolic deduction methods using non-classical mechanisms for producing plausible conclusions in spite of the partial lack of information (defeasible reasoning).

2.2. Imprecision

Imprecision refers to the contents of the considered statement and depends on the granularity of the language used to describe the information. For instance, the sentence "Paul is between 25 and 30 years old" is clearly imprecise, but the sentence "Paul is 26 years old" is precise if we only expect integer values for the age and is imprecise if the values should also indicate the number of months. Imprecise statements may stem from disjunctive information such as "Peter is 25 or 27", or negative information (when the underlying domain contains more than two values) such as "Peter is not between 25 and 29 years old". An extreme situation is when Peter's age is completely unknown which means that any value of the universe may equally be assigned. Imprecision is represented in terms of subsets of the relevant attribute domain which are not singletons. These subsets constrain the possible values which can be assigned to the attribute for the considered object.

A given statement may be both uncertain and imprecise. For instance, "Paul is 35 or 36 or 37 years old" is clearly imprecise, but in addition, we may have some doubt about our source of information and it becomes then uncertain. Generally, a balance exists between uncertainty and imprecision : "the more imprecise you are, the more certain your statement", and conversely "the more precise you are, the more uncertain the statement" with respect to a given state of available information. For instance if somebody, after meeting Peter for just a few minutes, says that "Peter is between 30 and 50", this can be regarded as certain or at least as almost certain, since it is a very imprecise statement for which there is almost no risk to be denied ; on the contrary if the person claims that "Peter is 35, 36 or 37 years old" this is more precise but the uncertainty should have increased.

Imprecision may also be due in a multiple source data base system to the use of different vocabularies for expressing attribute values, corresponding to different partitions of the same universe of discourse. In that case, there is not a one-to-one correspondence between the labels of the elements of the different partitions. Given two partitions $\mathcal{P}_1 = (A_1, \dots, A_n)$ and $\mathcal{P}_2 = (B_1, \dots, B_m)$ of a domain D , we can only define an upper approximation A_i^* and a lower approximation A_i_* of the element A_i of \mathcal{P}_1 in terms of those of \mathcal{P}_2 , namely $A_i^* = \{\bigcup B_j, A_i \cap B_j \neq \emptyset\}$, $A_i_* = \{\bigcup B_j, B_j \subseteq A_i\}$ thus giving birth to a rough set (Pawlak, 1991).

2.3. Vagueness and Gradual Properties

A vague statement contains vague or gradual predicates. It may also include vague quantifiers. For instance, "Paul is a young researcher" refers to Paul's age using the linguistic term "young". Note that the meaning of a vague predicate depends on the context : a big butterfly is smaller than a small elephant ; the context may even depend on the user. So, it is not a universally accepted meaning of "young" in a given context that we try to represent in practice, but the intended meaning of a given person. Fortunately, fuzzy sets are rather easy to elicitate (since it is sufficient in practice to identify

its support (the elements with non-zero membership) and its core (the elements with membership 1). See 4.1.3. A statement such as "Paul is young" is not necessarily true or false, and as already said, it may be used in two completely different situations : i) Paul's age is precisely known, for instance he is 30 years old, and the statement then receives a degree of truth which estimates to what extent 30 matches the representation of "young" in the considered context ; or ii) the fact that "Paul is young" is the *only available* information about Paul's age and thus the statement represents a flexible constraint on the acceptable values for Paul's age. In both cases, an ordering among the values compatible with the idea of "young" is defined.

Remark : It is possible to encounter second order imprecision and uncertainty if the value of a degree of truth or of a measure of uncertainty (probability, possibility, etc.) is imprecisely, vaguely stated, or not known with complete certainty. For instance, we may want to express that "it is not completely certain that John's is ill-paid". It gives birth to a possibility distribution of the type pictured in Figure 2. Such a possibility distribution expresses that there is a possibility equal to $1 - \alpha$ that John is not-ill paid and thus that "John is ill-paid" is only certain at degree α .

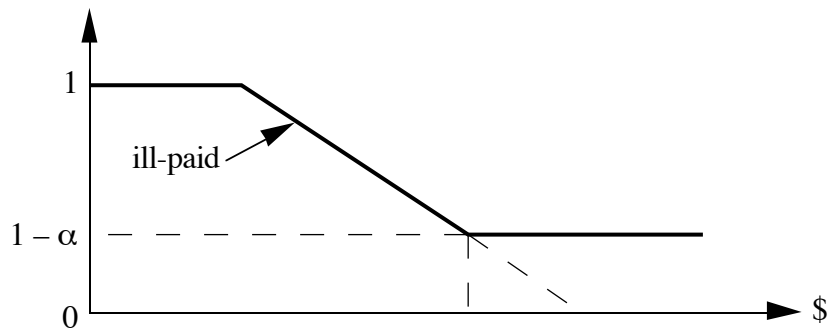


Figure 2 : Representing a vague and uncertain piece of information

2.4. Inconsistency

An inconsistent state of information is such that there is no possible assignment of the variable under consideration (or no model). The two pieces of information "Peter is 25" and "Peter is older than 27" illustrate this situation since there is no way of assigning a value to Peter's age which agrees with both pieces of information. This may be due to the existence of distinct sources of information with different levels of reliability. Acknowledging this fact may lead to a solution ; see Dubois, Lang and Prade (1992) where a set of n certainty degrees associated with the n sources under consideration is attached to each proposition in a possibilistic logic framework.

3. What 'Fuzzy Data Base' Means ?

The expressions "fuzzy data base" or "fuzzy data base system" have been used as generic terms for referring to any fuzzy set-based approach in information management. So these terms cover quite different views which are briefly reviewed in the following. Most of the approaches take place in the framework of the relational model of data, although the handling of fuzzy information in the entity-relationship model has been also explored (Zvieli and Chen, 1986 ; Ruspini, 1987).

If we take apart the idea of a fuzzy or flexible query (e.g., Tahani, 1977 ; Zemankova and Kandel, 1984 ; Dubois and Prade, 1988a ; Bosc et al., 1988) which is discussed in the next section, "fuzzy data base" may have at least four different intended meanings according to the way we use fuzzy sets for representing data pervaded with uncertainty and vagueness, or for modelling fuzzy concepts.

A first interpretation consists in extending the data base relational scheme by dealing with fuzzy relations rather than with ordinary relations, a fuzzy relation being defined as a fuzzy subset of

a Cartesian product of domains. Each tuple is provided with a grade of membership expressing the extent to which it belongs to the fuzzy relation. Such fuzzy relations made of weighted tuples can be either used as a format for storing information (Kunii, 1976 ; Baldwin and Zhou, 1984 ; Li and Liu, 1990) or only produced when answering flexible queries addressed to a non-fuzzy data base (Tahani, 1977 ; Bosc et al., 1988). In this latter case the relation represents a fuzzy concept, e.g. "employees who are middle-aged and recently-engaged" ; in such an example the weight attached to a tuple represents to what extent the corresponding employee is "middle-aged and recently-engaged". In this case the data are neither imprecise nor uncertain but a level of compatibility with respect to some fuzzy concept(s) of interest is associated to them.

However, relations with weighted tuples, which can be seen as fuzzy relations from a formal point of view, can be understood at least in two different ways depending on the intended meaning of the weights. Unfortunately this meaning remains sometimes unclear in the fuzzy set literature. Weights may represent : i) the compatibility of tuples with respect to a fuzzy concept as said above ; ii) the certainty we have in the information stored in the tuple, i.e., the information contained in the tuple is qualified with a certainty equal to the weight (we may also think of weighting the tuples in terms of possibility). This applies as well to relations modelling the association between values, as for instance in the fuzzy relation "likes (person, movie)", which can be either weighted in terms of uncertainty or in terms of strength of feeling. Note that if the weight models uncertainty, "likes" may be considered as an ordinary predicate, while if it models strength or intensity, "likes" is necessarily viewed as a fuzzy predicate (in such a situation the fuzzy relation represents the extension of a n-ary fuzzy predicate). In any case, weights can be assessed in terms of linguistic values (e.g. 'moderately', 'very unlikely'). Clearly the interpretation of the weights should be taken into account when defining operations for manipulating them.

A second view of fuzziness in data bases relies on the notion of interchangeability or tolerance relations defined on attribute domains (Buckles and Petry, 1982, 1987 ; Cayrol et al., 1982 ; Potoczny, 1984 ; McClelland et al., 1988 ; Zemankova, 1989 ; Rundensteiner et al., 1989). For instance, this approach may be used to represent data issued from some measurement devices in order to represent the error on the measure. Indeed, given an attribute domain \mathcal{D} , an interchangeability relation T may be defined through its membership function μ_T from $\mathcal{D} \times \mathcal{D}$ to $[0,1]$, such that the closer (or the more interchangeable) the values d and d' , the closer to 1 the degree of membership $\mu_T(d,d')$. The value of μ_T may be explicitly given by an expert for each pair (d,d') for discrete domains, if no distance is available, or μ_T may be built from a distance, especially on continuums (in practice, closed sub-intervals of the real line). Given a distance δ , we may for instance define μ_T as : $\max(0, 1 - \delta(d,d')/\lambda)$ where λ is a positive real number. The interchangeability relation attached to each attribute domain may be understood as a standard amount of fuzziness which should be attached to each (apparently) precise value of this attribute stored in the data base in order to restore the inherent imprecision or uncertainty of this value (Prade and Testemale, 1987b). In this case we acknowledge the fact that there is an approximate "synonymy" between close elements in attribute domains and that when a precise attribute value is stored, another value, close to the former value, might have been stored as well. Another use of fuzzy relations expressing proximity is when a tolerance is added to a request in order to enlarge the set of acceptable values in a fuzzy way.

A third type of fuzzy data base that we shall consider in greater details in Section 5 allows for the representation of ill-known attribute values inside tuples by means of possibility distributions (Zadeh, 1978). For instance, in a database describing houses, if the size (or the price) is not perfectly known, it will be described by means of a possibility distribution (which in fact restricts the set of possible values for the size (or the price) of a particular house). This approach has been particularly developed by Umano (1982, 1983), Umano and Fukami (1987), Prade, (1982, 1984), Prade and Testemale (1984), Zemankova and Kandel (1984), Vandenberghe et al. (1989). In this approach, attribute values are allowed to be precise, imprecise, vague, pervaded with uncertainty, unknown, or inapplicable, according to the nature of the available information. In all these cases, the information is represented by means of a possibility distribution $\pi_{A(x)}$ which restricts the possible values of the attribute A for the considered object x . $\pi_{A(x)}$ takes its values in the interval $[0,1]$, $\pi_{A(x)}(u) = 0$

means that u is impossible and thus totally excluded for $A(x)$, while $\pi_{A(x)}(u) = 1$ means that u is completely possible for u ; clearly several distinct values can be considered as completely possible for $A(x)$ in case of imprecise information. The values restricted by $\pi_{A(x)}$ are *mutually exclusive* as possible values of $A(x)$, and thus $\pi_{A(x)}$ is defined on D if A is single-valued, and on 2^D if A is an attribute which can have several values in D simultaneously (e.g., the language(s) spoken by a person).

A precise, imprecise or vague piece of information can be qualified in terms of uncertainty. As already said, one interpretation of a weighted tuple is to view the weight attached to the tuple as an estimate of the reliability, or of the certainty of the information contained in the tuple. Modelling this certainty in terms of a necessity measure in the framework of possibility theory leads to associate the level of certainty attached to the tuple to *each* of the elements of the tuple and to modify each possibility distribution in the tuple into another possibility distribution reflecting this level of certainty. Figure 2 gives an example of how a possibility distribution, here representing the fuzzy set of possible values compatible with the idea of "ill-paid", is modified into another possibility distribution to acknowledge the fact that the information is only certain at the degree α .

Note that in the above-mentioned representation of pieces of information by means of possibility distributions, the values which are outside the fuzzy set whose membership function is equal to the possibility distribution, are definitely excluded as possible values (since their possibility degree is zero). In other words, we are totally *certain* that the value of the attribute for the considered item is among the values with a non-zero degree of possibility. However we may imagine as suggested in Dubois and Prade (1991) to have to store in a database a value or a set of values which are individually judged as more or less possible for the considered attribute of the object under concern, without completely knowing the set of all its possible values. In this case we are no longer certain that the value of the attribute is among the values stored in the databases, we only know that these values are possibilities for sure (but there may be other ones as well). This corresponds to possibility-qualified pieces of information and may be considered as a fourth kind of fuzziness in databases. If more information of this kind is provided it would lead to enlarge the scope of possibilities, while in the case of a possibility distribution *restricting with certainty* the possible values, the scope of possible values can only diminish with the arrival of new information. In case of possibility-qualified pieces of information, the intended purpose is to express that there is a "guaranteed possibility" for a given attribute value to apply to the considered item. For instance, when we have examples of *possible* prices for a second-hand car (these prices do not necessarily intend to describe all the more or less possible prices with certainty).

Apart from the relational view of a database expressed in terms of values of attributes in tuples, we may use a logical framework for the representation of data (and integrity constraints) in deductive databases. Then treatment of uncertainty in the setting of possibility theory can be handled in possibilistic logic (Dubois, Lang and Prade, 1989, 1991, 1993); the case of flexible queries would require the use of fuzzy predicates. The reader is referred to the above-mentioned references for introductions to this approach, which will not be recalled in the following for the sake of brevity. See also Steger et al. (1989) for an example of handling of certainty degrees using fuzzy set combination operations in a logic programming style.

Remark : Flexible Querying and Information Retrieval

In the context of information retrieval systems, flexible querying makes sense since by nature the result should be an ordered list of documents matching the user's needs. Then it may seem natural to attach a degree of importance to each key-word appearing in a query to make it flexible by expressing preferences, and/or to grade the relevance of a key-word with respect to a document in order to take into account the uncertainty pertaining to the appropriateness of the keyword for the document (Negoiita and Flondor, 1976; Bollman and Konrad, 1976; Bookstein, 1980; Kohout et al., 1984; Kraft and Buell, 1983; Yager, 1987; Biswas et al., 1987; Murai et al., 1989; Miyamoto, 1990; Nomoto et al., 1990; Ogawa et al., 1991; Bordogna et al., 1991). Fuzzy thesauri taking into account the approximate synonymy between key-words can be also introduced (e.g., Reisinger, 1974).

4. Flexible Queries

When querying a database, one sometimes does not wish to define precise limits of acceptance or rejection for a condition, especially when the satisfaction of this condition is a matter of degree. This also occurs when one wishes to express preferences and thus to distinguish between acceptable and non-acceptable items in a more refined way than with a Boolean filter. In order to do that, vague predicates are allowed in the requests ; such predicates are represented by fuzzy sets and model gradual properties whose satisfaction is a matter of degree ; we can thus express that among acceptable attribute values, some are preferred. A query looking for "hotels not too expensive close to downtown" illustrates this expression of preferences. An advantage of flexible queries is first to provide a reply when a classical crisp, too requiring request would have produced an empty response, and secondly to rank-order the n best replies rather than to provide a long list of undifferentiated replies to the user. An advantage of this approach is that the user has not to reformulate his request too many times.

4.1. The Fuzzy Set Approach

4.1.1. Overview

Here, vague criteria involved in requests are assumed to be represented by fuzzy sets. A query will be constructed using several types of vague predicate expressions :

- atomic predicate : defined by a fuzzy set on a domain D_i , i.e., by a membership function from D_i to $[0,1]$; examples of such vague predicates are "tall", "young", "important", etc., to be represented in a given context. In practice, unimodal functions with a trapezoidal shape are often used.
- modified predicate : modifier functions, from $[0,1]$ to $[0,1]$, can be applied to fuzzy set membership functions in order to model the effect of linguistic hedges such as "very", "more or less", "rather", etc. The most usual modifier functions are of the form $y = x^\alpha$, or are translating operations. Antonyms can also be treated in this category : the antonym $\text{ant}(T)$ of a fuzzy label T is defined by taking the membership function of $\text{ant}(T)$ as the symmetrical of the one of T with respect to the axis passing by the middle of the domain D on which these functions are defined, namely if $D = [a,b]$, $\mu_{\text{ant}(T)}(x) = \mu_T(a + b - x)$. Thus $\text{ant}(\text{ant}(T)) = T$.
- compound predicate : it is defined as a combination of membership functions representing atomic predicates by means of fuzzy set connectives ; for instance we may have a request for "cars not expensive and rather recent (or with a limited mileage)" ; see Section 4.1.2.

Besides queries may involve vague quantifiers (e.g. "most", "a few", "about a dozen"). It may be queries about the (relative) cardinalities of some subsets of objects, as for instance in "Is it true that most of the employees which satisfy A satisfy also B ?", or even queries looking for objects satisfying *most* of the prescribed conditions.

Queries which look for the maximal, minimal or average values, or more generally for the evaluation of any scalar function, over a set objects specified in a fuzzy way can also be handled. An example of such a query is "what is the average salary of *young* employees ?". See Dubois and Prade (1990) for a presentation of scalar and fuzzy-valued solutions.

4.1.2. Connectives

Compound conditions expressed under the form of logical expressions are represented using fuzzy set operations. Fuzzy set theory offers a panoply of aggregation attitudes richer than the Boolean framework. We have at our disposal a variety of conjunctions and disjunctions, but also trade-off operations. Let $A_i(x)$ be the precise value of the attribute A_i for the item x and P_i be the subset expressing the restriction for $A_i(x)$, in the request. Thus, conjunctive (resp. disjunctive)

aggregations of the elementary degrees of matching $\mu_{P_i}(A_i(x))$ are performed usually by applying min (resp. max) operation to the degrees. Using min for evaluating a conjunction of required properties means that the grading of the least satisfied property will reflect the global level of satisfaction. This type of logical conjunction may be felt as too requiring in some situations. See below for other possible candidates for modelling a "and" connective. Negation is modelled by $1 - \mu_{P_i}(A_i(x))$ which represents the extent to which $A_i(x)$ belongs to the complement of P_i . In some applications, we may like to express that some elementary conditions are less important than others. Conjunctive and disjunctive aggregations are respectively generalized in that case by (Dubois and Prade, 1986 ; Sanchez, 1989)

$$\min_i \max(\mu_{P_i}(A_i(x)), 1 - w_i) \\ \max_i \min(\mu_{P_i}(A_i(x)), w_i)$$

where w_i is a weight of importance of the condition bearing on the attribute A_i in the request. The weights are supposed to satisfy the normalization condition $\max_i w_i = 1$. Clearly, when all the elementary conditions are equally important, (i.e., $\forall i, w_i = 1$), the two operations above reduce respectively to min and max. When $w_i = 0$, no condition on the attribute A_i is taken into account. We observe, in the case of the conjunctive combination, that even if $A_i(x)$ fails to satisfy the restriction P_i of importance w_i , this cannot make the global result of the combination going below $1 - w_i$. For instance, if we look for an apartment which is "cheap and sufficiently large" (the second criterion being less important, we are led to evaluate an expression of the form $\min(\mu_{\text{cheap}}(\text{price}), \max(\mu_{\text{sufficiently-large}}(\text{surface}), 1 - w))$. In this expression, if the surface is not sufficiently large, the global rating will be equal to $1 - w$ rather than to zero if the price is OK. Thus, an apartment cannot be completely rejected only on the basis of its surface in this case.

However, conjunction and disjunction operations, other than min and max, can be used. There exist more drastic conjunction operations (e.g., the product $a \cdot b$, or $\max(0, a + b - 1)$) and less drastic disjunction operations (e.g., the "probabilistic sum" $a + b - a \cdot b$, or the bounded sum $\min(1, a + b)$). There also exist many trade-off operations between min and max (e.g., the arithmetic mean) which can model compensatory "and" for instance (a low degree of satisfaction for one elementary condition can be somewhat balanced by a high degree of satisfaction for another condition). See Zimmermann and Zysno (1980) for instance.

Overviews on fuzzy set aggregation connectives can be found in (Dubois and Prade, 1985 ; Yager, 1991a). See also Su et al. (1987) for an applied perspective based on "an extended continuous logic" (in fact on fuzzy sets). Let us also mention ordered weighted average (OWA) operations, defined by $\text{OWA}(a_1, \dots, a_n) = w_1 b_1 + \dots + w_n b_n$ where b_i denotes the i^{th} largest value among the a_i 's and the sum of the w_i 's equals 1 ; this aggregation performs a somewhat dynamic weighting (the weights apply after ordering the values to be aggregated and do not depend on the places of these values in the initial list), which contrasts with the ordinary weighted mean (Yager, 1988).

4.1.3. Elicitation of Membership Functions and Connectives

An important issue is the elicitation of the fuzzy set membership functions representing the fuzzy predicates or fuzzy predicate expressions involved in a user query. There are two main possibilities. A first solution consists in the use of a standard vocabulary with a set of connectives (and possibly of fuzzy set modifiers) made available to the user for specifying requests ; this requires that the user is familiar with the intended meaning of the fuzzy vocabulary. A better solution, in general, seems to have a procedure which elicits the intended meaning of user's words. Indeed what is of interest is what the user means by "not-too-expensive" for instance and not to have a universally valid representation of "expensive" and of the modifier "not-too", which may not exist. The procedure first elicits the membership functions of the fuzzy labels referring to a unique domain D_i , mainly by identifying when this function should be zero and when it is equal to 1 and by

providing gradual transition(s) elsewhere. Then, we have to identify the intended meaning of the connectives used in the query in case of compound predicates. Indeed a conjunction like "and" may have a variety of different meanings ranging from a purely logical interpretation (requiring the more or less complete satisfaction of each of the conditions involved in the conjunction) to compensatory "and" 's allowing for trade-offs between the levels of satisfaction of the conditions. A simple procedure for the elicitation of connectives is presented in (Dubois and Prade, 1988a, Chapter 3). It requires global ratings given by the user for few prototypical objects.

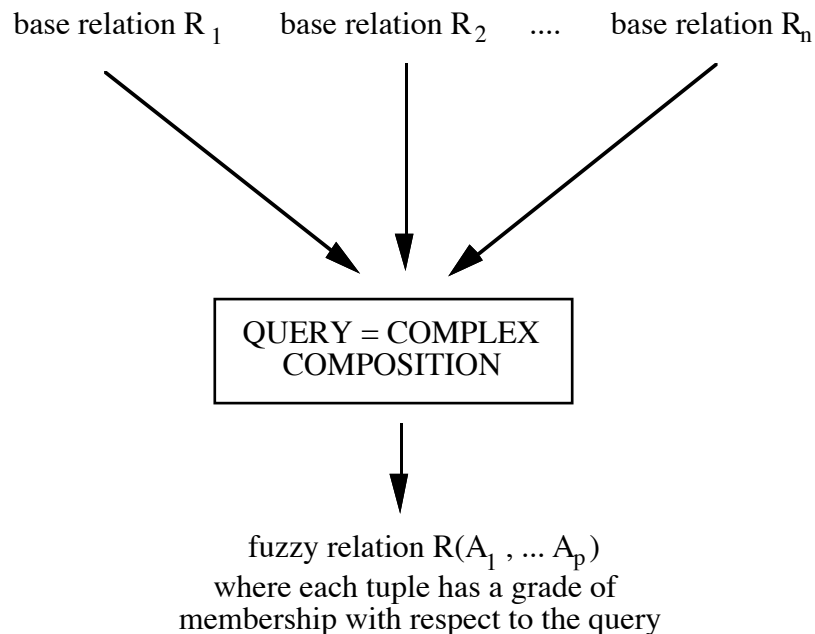
As far as we are using max or min operations and the complementation to 1 (which is an order-reversing operation), it can be shown that computation is not sensitive to slight variations of the membership grades, since these operations are concerned only by the ordering between the degrees. Thus, a very precise identification of the membership functions is not necessary in practice. However, it is important to be aware that fuzzy sets require a commensurability hypothesis, i.e., the use of a common scale for grading the compatibility with respect to the different predicates.

4.2. Flexible Querying of Regular Data Bases

In this section, we consider usual databases where data are assumed to be precisely known and we investigate the issue of flexible querying. First of all, we briefly review the approaches suggested in the literature and their position with respect to fuzzy sets, then some elements of an extended SQL-like language are given. Finally, query processing aspects are discussed.

4.2.1. Fuzzy Set Based Approach

From an algebraic point of view, we can consider that the relational operations (selection, projection and join) and set operations are extended to fuzzy relations according to the following schema :



One can thereby select one relation or the product of several relations and receive a projection of all the tuples provided with a degree of membership with respect to the query. Answers are then ordered and this provides a way for avoiding (at least partly) empty or too large set of answers.

4.2.2. Extensions of Relational Systems

Other approaches to flexible querying have been proposed and can be divided into three groups : i) introduction of a complementary criterion in the query, ii) use of similarities and distances and iii) description of explicit preferences and weighting.

Complementary Criterion

In the PREFERENCES system (Lacroix and Lavency, 1987), a question is composed of a principal condition C and a complementary part P that is relative to the description of preferences, both of which are based on Boolean expressions. The meaning of this type of question is : "find the tuples which satisfy C and rank them according to their satisfaction of P ." This system allows for the combination of preference clauses (P) by means of two constructors : nesting (hierarchy of conditions) and juxtaposition (conditions of equal importance). From the subset R_C of the tuples of a relation R satisfying condition C , the nesting (resp. juxtaposition) of preference clauses P_1, \dots, P_n leads to the sets : S_1 the subset of R_C satisfying P_1 and not P_2 (resp. one single clause); S_2 the subset of R_C satisfying P_1 and P_2 but not P_3 (resp. exactly two clauses) ; ... ; S_n the subset of R_C satisfying P_1 and ... and P_n . The user receives as an initial response the set S_i , with the highest index i which is not empty, and he can go back to the previous sets, all of which corresponds to a weakening of the preference condition.

One of the significant advantages of this system is the avoidance of successive formulations in reaching a desired set of responses. The authors rightly point out that to handle an equivalent formulation in a classic language would be rather tiresome. However, it must be noticed that the discrimination capacity remains limited, since it directly depends on the number of preference predicates given by the user. If n such predicates are provided (in practice n is small - less than 10), any tuple belongs to one of n possible classes.

Another attempt based on a complementary criterion has been suggested with the system called DEDUCE2 (Chang, 1982). However, it has been shown (Bosc et al., 1992a) that the composition of predicates, which is based on ranks issued from sorts, is not semantically founded.

Similarities and Distances

Another idea relies upon queries which include conditions based on the notion of similarity (denoted here by \approx) rather than strict equality. Let us consider conditions of the type " $A \approx v$ " where " v " represents an ideal value, but where other values are nevertheless acceptable ; for instance salary $\approx \$2000$ means that \$2000 is excellent but values around (the interval [1950-2050] for instance) can also be accepted. The evaluation of such a condition on an element x is made using a distance according to the following principle : if $A(x)$ is somewhat similar to the value " v ", then the dissimilarity between x and the ideal value will be estimated by the distance between $A(x)$ and v , otherwise the dissimilarity will be infinite. In the presence of connectors such as conjunction and disjunction, an overall distance must then be calculated, thus allowing the elements concerned to be ordered. Several systems based either on an explicit operator of similarity (ARES (Ichikawa and Hirakawa, 1986), VAGUE (Motro, 1988)), or an implicit one (the "nearest neighbors" technique proposed by (Friedman et al., 1975)), have been suggested.

In ARES, elementary distances are attached to a given domain and are given by means of a relation expressing the distance between any two values. In a given query (which involves both Boolean predicates and predicates involving similarity that can only be ANDed), the user chooses a threshold (t_i) for each predicate involving a similarity. The global distance is defined as the sum of the elementary distances tied to the similarity predicates involved in the query.

In VAGUE, we have three main differences with respect to ARES : the disjunction of predicates is allowed, similarity predicates can be explicitly weighted and the global distance

mechanism for a conjunction is based on the euclidian distance. ARES and VAGUE share a common characteristic, namely the fact that only elements which more or less satisfy any term of a conjunction are ordered. It is clear that this behaviour may cause some sort of discontinuity since an element which is satisfying very poorly each term will be retained whereas one satisfying very well all terms but one, will be definitely rejected. It is to notice that this drawback can be obviated in a fuzzy set oriented environment.

In the "nearest neighbors" approach a query involves a set of values which characterize an ideal tuple X . Each concerned tuple is then compared with X by means of a global function which gathers the results of local distance functions applied to some attributes. One of the most used global functions is the L_p -norm defined as :

$$\sqrt[p]{\sum_{i=1}^n \text{dist}_i(x)^p} \quad \text{with } \text{dist}_i(x) = \frac{|x_i - X_i|}{\max_i - \min_i}$$

where x_i and X_i stand for the values of the i^{th} attribute of the current tuple and the prototype, which can vary between \min_i and \max_i .

Criteria with Preferences and Weighting

In the framework of information retrieval, the flexible retrieval system called MULTOS (Rabitti, 1990) has been proposed. Its principle consists in replacing a traditional criterion with a set of criteria to which an explicit preference level (expressed by a value between 0 and 1, or by a linguistic term) is attached. Thus, if we are interested in the year of publication, we might write : $\{\text{year} \in [1978, 1982] \text{ "preferred"}, \text{year} \in [1983, 1988] \text{ "accepted"}\}$. Furthermore, we can weight each set of criteria (e.g. the subject matter is more important (high) than the price of the document (medium) which in turn is more important than the year of publication (low)). The conjunction and disjunction connectives allow the combination of several criteria in the queries. In particular, the semantics of the conjunction is expressed by the weighted sum (the aforementioned linguistic values "high", "medium" and "low" being encoded by values between 0 and 1) of the results weighted according to the preference attached to them (accepted, preferred).

The analysis of these three kinds of non fuzzy approaches aiming at discriminated answers (Bosc and Pivert, 1992a, b) shows that in any case : i) any of these queries is expressible in the fuzzy sets framework, ii) the ordering mechanism is basically a mean, iii) the allowed queries have a very special typology and iv) each system proposes only one (or two) aggregation mechanism(s) and it is then clear that fuzzy sets provide a much more general framework for choosing and expressing the appropriate aggregation mechanism.

4.2.3. Survey of an Extended SQL-Like Language

Query languages of the SQL type are very popular and represent a standard. This motivates an extension of SQL into a language SQL_f allowing the processing of fuzzy queries (Bosc et al., 1988 ; Bosc and Pivert, 1991). The basic block in SQL is :

select attributes **from** relations **where** condition

The basic idea consists in introducing imprecise predicates into the condition (Boolean conditions being a particular case). Furthermore, in order to ensure the calibration of the number of responses, a number of desired responses (quantitative aspect) and/or the minimum degree of satisfaction required for selection (qualitative aspect) is specified. The result of such a query is then a fuzzy relation.

Elementary predicates (unary or n-ary, including (fuzzy) relational comparators) allow the comparison between an attribute and a given value or between two attributes. Modified predicates thanks to modifiers such as "very", "fairly", "relatively" are also allowed. With the help of these two types of basic predicates, compound predicates using connectives can be defined. Conjunction and disjunction (binary or n-ary), defined in terms of intersection and union of fuzzy sets, generalize the Boolean AND and OR, and it is possible to define other aggregation operators such as the means (arithmetic, geometric, weighted) which convey compensation effects.

In SQL, it is possible to nest blocks using operators such as "in" or "exists" and it is useful to study how these constructions can be extended. This point is all the more important as it poses the question of preserving the equivalences of different expressions of the same query. Below we give as an illustration the expression in four different forms of a question in standard SQL and the corresponding forms when fuzzy conditions are introduced, which remain equivalent as in the non-fuzzy case. We consider a database made of the two relations "employee" and "department", their format being :

Emp (#emp, name, #dept, salary, job), Dept (#dept, budget)

"number and name of any employee working in a department whose budget is 1000 times the employee's salary "

a) **select** #emp, name **from** Emp, Dept **where** Dept.#dept = Emp.#dept **and** Dept.budget = Emp.salary * 1000

b) **select** #emp, name **from** Emp **where** #dept **in** (**select** #dept **from** Dept **where** budget = Emp.salary * 1000)

c) **select** #emp, name **from** Emp **where** (salary * 1000) **in** (**select** budget **from** Dept **where** #dept = Emp.#dept)

d) **select** #emp, name **from** Emp **where** **exists** (**select** * **from** Dept **where** #dept = Emp.#dept **and** budget = Emp.salary * 1000)

"number and name of any employee working in a department whose budget is about 1000 times the employee's salary "

a') **select** #emp, name **from** Emp, Dept **where** Dept.#dept = Emp.#dept **and** Dept.budget \approx Emp.salary * 1000

b') **select** #emp, name **from** Emp **where** #dept **in** (**select** #dept **from** Dept **where** budget \approx Emp.salary * 1000)

c') **select** #emp, name **from** Emp **where** (salary * 1000) **inf** (**select** budget **from** Dept **where** #dept = Emp.#dept)

d') **select** #emp, name **from** Emp **where** **exists** (**select** * **from** Dept **where** #dept = Emp.#dept **and** budget \approx Emp.salary * 1000)

The operator of comparison \approx means "approximately equal to". The operators "in", "inf" and "exists" (in the right hand column) respectively correspond to the membership of an element to a fuzzy set, the fuzzy membership of an element to a fuzzy set and the existence of an element in a fuzzy set. It is possible to show that, if appropriate definitions are chosen for the extended nesting operators, then expressions a'), b'), c') and d') deliver the same result as a), b), c) and d) do.

It is also possible to introduce fuzzy predicates operating no longer on individual tuples, but on sets of tuples by extending the framework of the partitioning offered by SQL. On this level, we find two main types of imprecise condition : i) a fuzzy predicate is applied to the result of a function aggregating tuples (max, sum, mean, for example), ii) fuzzy quantifiers are brought into play. Thus, the request "find the best 10 departments such that the average salary of the secretaries should be around \$6000" will be expressed by :

select 10 #dept **from** Emp
where job = "secretary"
group by #dept **having** avg(salary) \approx 6000

and the query "find the 10 best departments such that most of the young employees are well-paid" will be expressed by :

```
select 10 #dept from Emp group by #dept
having most_of (young are well-paid)
```

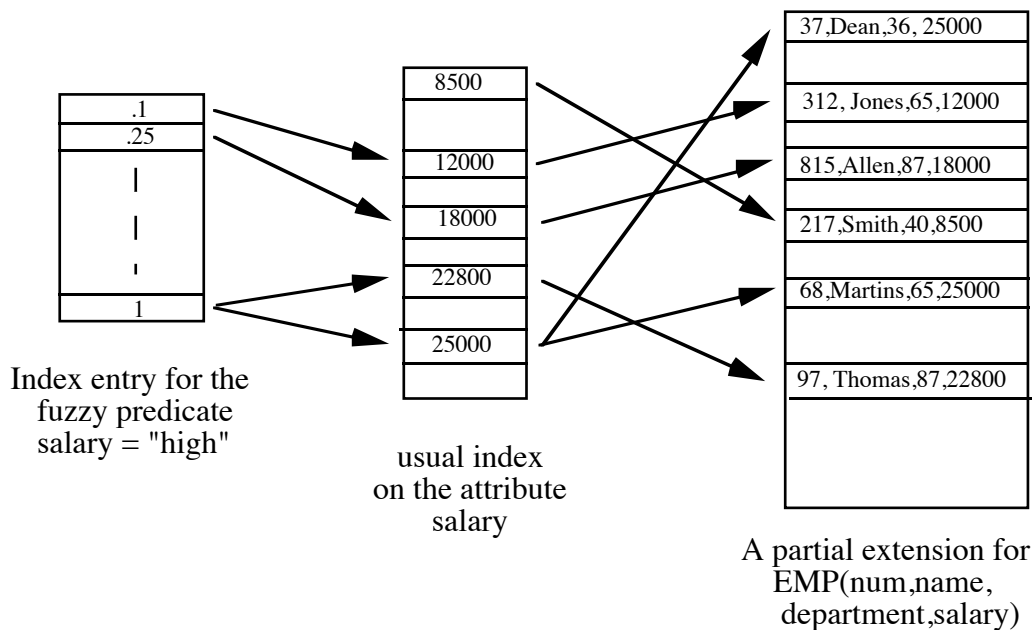
See Bosc and Pivert (1992c), Kacprzyk et al. (1989) for requests involving fuzzy quantifiers. See also Yager (1991b) for the definitions of fuzzy quotient operators from fuzzy quantifiers and OWA operations.

4.2.4. Query Processing

The evaluation of queries in the context of declarative languages of the relational type, for example, remains at the present time, in spite of numerous works of research, an open problem. Indeed, nobody knows how to find the best plan of execution (algorithm) of any request in a "reasonable" time. Based on this statement, it is clear that we cannot hope to achieve an optimal response to the evaluation of imprecise requests inasmuch as these requests are themselves at least as complex as ordinary ones. The increase in complexity is due to two main causes : i) the usual access paths such as index are no longer usable, ii) the operations to be carried out depend on a larger volume of data since the selections, joins, etc. produce more tuples than in the Boolean context. In what follows, we give some elements relative on the one hand to the definition of index usable for fuzzy predicates and, on the other hand, to an approach called derivation which allows the evaluation of the relational queries which requires only one projection, some selections and some joins.

Fuzzy Predicates Indexing

In a classic DBMS, indexation gives direct access to the tuples which have a given value of the indexed attribute. This type of structure is unfortunately only usable if the requests contain classic predicates where a value (or an interval) is referred to. In order to take into account imprecise predicates, we can construct an index for a given fuzzy predicate which, representing a degree of membership at each input, associates the tuples satisfying this predicate with the degree under consideration. This process can use an already existing index according to the figure below :



This kind of structure provides an efficient access to tuples as far as the degree is known.

Derivation Method

For the sake of simplicity, we consider here compound fuzzy predicates applied to a single relation and their λ -level cuts, i.e., the elements (tuples) satisfying these predicates at a level higher than or equal to λ . The idea is to derive one (or several) Boolean necessary condition(s) for the belonging of an element to a λ -cut. Consequently, we can evaluate this condition using a normal DBMS in order to construct a subset (with cardinality much lower than that of the original relationship) to which the original fuzzy predicate will then be applied. In some cases, necessary and sufficient conditions are found, which permits the direct construction of the desired λ -cut. The problem to be solved is the distribution of the λ -cut operation on a compound predicate.

If the fuzzy predicate under consideration is atomic and represented by a trapezoid (a common case in practice), the λ -cut corresponds to the membership of the variable to an interval. Thus we obtain a simple Boolean condition which is both necessary and sufficient, in the form $x \in [v_1, v_2]$. If a linguistic modifier is used, it is easy to show that we are back to a condition of the same type. In the case of compound predicates, the idea is to express the λ -cut of the compound predicate in terms of λ -cuts bearing on the constituting predicates (and therefore expressible as condition of the form $x \in [v_1, v_2]$). Here, two types of situation may occur (Bosc and Pivert, 1992d) : i) it is possible to find out an equivalent formula, or ii) it is only possible to exhibit an expression whose result is a superset of the elements satisfying the initial λ -cut. A large number of connectors have been examined and necessary conditions have been provided for each of them. Once the final expression got, the idea is to submit it to the regular DBMS (which should be able to process it efficiently) in order to get a "reasonable" subset of tuples against which the initial fuzzy query will be processed.

A Strategy for Processing Quantified Queries

Now, we would like to take into account queries where a condition applies to a set of tuples and not to individual tuples as for the derivation method. We point out a strategy (Bosc and Pivert, 1992c) whose goal is essentially to avoid the exhaustive scan of the elements of a set, thus likely to save disk accesses which represent the main cost for the database query evaluation process.

The two following queries are illustrations of the use of quantified propositions in the scope of database querying :

- 1) find the best 10 departments where at least three employees are "middle-aged"

```
select 10 dep from EMPLOYEE
group by dep having at least three are middle-aged
```

- 2) find the best 10 departments where almost all "low salary employees" are "recently engaged"

```
select 10 dep from EMPLOYEE
group by dep having almost-all low-salary are recently-engaged.
```

These two queries invoke the grouping mechanism of the SQL language.

A naive algorithm would be to access all the tuples of a given partition (having the same value for the considered attribute) and to compute the value of the quantified proposition. If this value is over a given threshold (λ -cut), the partition is retained and the next partition is in turn considered.

Another idea is to take advantage of properties of the quantifier (especially monotonicity if the quantifier is represented as an OWA aggregation) to improve the evaluation. This goal can be

achieved for queries of type 1) as far as the number of tuples of any partition can be known without scanning the partition. It is then possible to get conditions for partial evaluations of the OWA aggregation in order to decide after each new access to a tuple whether : i) it is sure that the desired degree (λ) will not be attained (failure condition), ii) this degree will be attained for sure (success condition) or iii) none of the previous conclusion holds and a new access is needed. In case i) or ii) it is possible to conclude about a partition before all its tuples have been accessed and thus data accesses are saved (Bosc et al., 1992c).

5. Imperfect Data in a Database

5.1. Representing Imperfect Data

Presently, available database systems only accept two types of values : precise values and null values of various kinds. The representation of disjunctive information and more generally of imprecise, vague or uncertain data in data bases has been investigated in logical (Demolombe, 1992) or in probabilistic frameworks (Wong, 1982 ; Barabará et al., 1992). See Zimanyi (1992) for a general overview encompassing both logical and numerical uncertainty formalisms.

In the possibility theory-based approach (Prade, 1982, 1984 ; Prade and Testemale, 1984, 1987), the available information about the value of a single-valued attribute A for a tuple x is represented by a possibility distribution $\pi_{A(x)}$ on $D \cup \{e\}$ where e is an extra-element which stands for the case when the attribute does not apply to x . The possibility distribution $\pi_{A(x)}$ can be viewed as a fuzzy restriction of the possible value of $A(x)$ and defines a mapping from $D \cup \{e\}$ to $[0,1]$. If information is consistent, there should exist a value in $D \cup \{e\}$ for $A(x)$, which leads to the normalization condition $\max_D \pi_{A(x)}(d) = 1$ (i.e., at least one value in $D \cup \{e\}$ is completely possible). For instance, the information "Paul is young" will be represented by : $\pi_{Age(Paul)}(e) = 0$ and $\pi_{Age(Paul)}(d) = \mu_{young}(d)$, $\forall d \in D$. Here, μ_{young} is a membership function which represents the vague predicate "young" in a given context. It is important to notice that the values restricted by a possibility distribution are considered as mutually exclusive. The degree $\pi_{A(x)}(d)$ rates the possibility that $d \in D$ is the right value of the attribute A for x . $\pi_{A(x)}(d) = 1$ only means that d is a completely possible value for $A(x)$, but does *not* mean that it is certain that d is the value of A for x (or in other words that d is necessarily the value of A for x), except if $\forall d' \neq d, \pi_{A(x)}(d') = 0$. Moreover the possibility distribution $\pi_{A(x)}$ is supposed to be normalized on $D \cup \{e\}$, i.e., $\exists d \in D$ such that $\pi_{A(x)}(d) = 1$ or $\pi_{A(x)}(e) = 1$, since either at least one value of the attribute domain is completely possible, or the attribute does not apply. This approach proposes a unified framework for representing precise, imprecise, as well as vague, values of attributes, and the following null value situations : i) the value of A for x is completely unknown : $\forall d \in D, \pi_{A(x)}(d) = 1, \pi_{A(x)}(e) = 0$, ii) the attribute A does not apply to x : $\forall d \in D, \pi_{A(x)}(d) = 0, \pi_{A(x)}(e) = 1$, and iii) we don't know whether we are in situation i or ii : $\forall d \in D, \pi_{A(x)}(d) = 1$, and $\pi_{A(x)}(e) = 1$. Multiple-valued attributes can be dealt with in this framework too.

The interest of such an approach is thus to be able to represent in a unified way, precise values as well as null values and ill-known values as shown on the following figures where different states of knowledge about John's salary are represented.

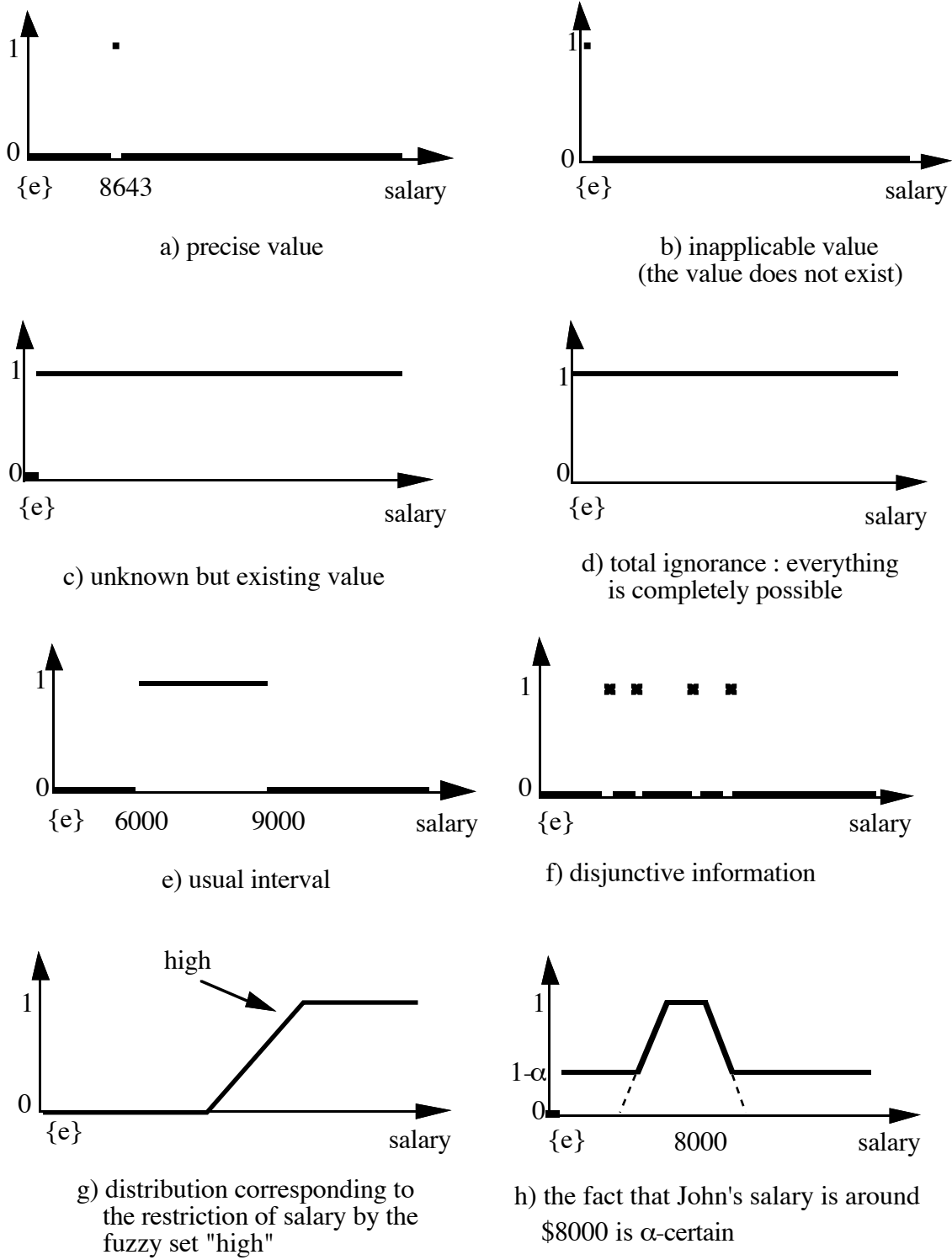


Figure 3 : Different kinds of incomplete information

In these examples, we have considered a single-valued attribute, but multiple-valued attributes can be handled as well. In our approach, multiple-valued attributes can be formally dealt with in the same manner as single-valued ones, provided that we work with possibility distributions defined on the power set of the attribute domains rather than on the attribute domains themselves. Indeed, in the case of multiple-valued attributes, the mutually exclusive possibilities are represented by subsets of values. Let us consider the example of languages spoken by an employee, the universe being {English, German, Spanish, Italian} and let us suppose that we know the following : we are certain that the employee Peter speaks English, it is totally possible that he also speaks German or Spanish,

Italian is unlikely and incompatible with Spanish. We therefore have the distribution (the possibility degree is given after the subset) :

$$\{\text{English}\}/1, \{\text{English, German}\}/1, \{\text{English, Spanish}\}/1, \{\text{English, German, Italian}\}/0.3, \\ \{\text{English, German, Spanish}\}/1, \{\text{English, Italian}\}/0.3.$$

See (Prade and Testemale, 1987) for details and (Dubois and Prade, 1989) for lower and upper approximations of possibility distributions on 2^D in terms of subsets of D . Relevant keywords for a document is a good example of a multiple-valued attribute ; see Prade and Testemale (1987c) for a treatment of keywords in this spirit. Another representation issue which can be dealt with in the possibilistic framework is the handling of interactivity constraints, e.g., "John and Paul have the same age which is between 20 and 30".

Clearly a similar approach can be developed in the probabilistic framework by using probability distributions in place of the possibility distributions $\pi_{A(x)}$. However possibility distributions, because of their "ordinal" nature, may be easier to elicitate. Also their normalization is easier than in the probabilistic case when we have not an exhaustive knowledge of the attribute domain. Moreover the probabilistic setting do not allow us to extend to gradual scales the modal distinction between what it is just possible and what is certain or necessarily true as recalled in Section 5.2. In other words, while $P(A) = 1 \Leftrightarrow P(\bar{A}) = 0$, we have for a possibility measure Π , $\Pi(A) = 0 \Rightarrow \Pi(\bar{A}) = 1$ but *not* $\Pi(\bar{A}) = 1 \Rightarrow \Pi(A) = 0$. This enables us to distinguish between the certainty that A is false ($\Pi(A) = 0$) and the total lack of certainty that A is true ($\Pi(\bar{A}) = 1$). Possibility theory is well-suited for modelling states of partial ignorance.

5.2. Matching Mechanism

When a condition applies to imperfectly known data, the result of a query evaluation can no longer be a single value. Since we do not know the precise values of some attributes for some items, we may be uncertain about the fact that these items satisfy or not the query (to some degree). It is why we use two degrees attached to two points of view : the extent to which it is possible that the condition is satisfied and the extent to which it is certain that the condition is satisfied. From the possibility distributions $\pi_{A(x)}$ and a subset P (ordinary or fuzzy), we can compute the fuzzy set ΠP (resp. NP) of the items whose A -value possibly (resp. necessarily) satisfies the condition P .

The membership degree of an item x to ΠP and NP are respectively given by (Dubois and Prade, 1988) :

$$\mu_{\Pi P}(x) = \Pi(P ; A(x)) = \sup_{d \in \mathcal{D}} \min(\mu_P(d), \pi_{A(x)}(d)) \\ \mu_{NP}(x) = N(P ; A(x)) = \inf_{d \in \mathcal{D} \cup \{e\}} \max(\mu_P(d), 1 - \pi_{A(x)}(d))$$

$\Pi(P ; A(x))$ estimates to what extent there is a value restricted by $\pi_{A(x)}$ compatible with P and $N(P ; A(x))$ to what extent all the values more or less possible for $A(x)$ are included in P . It can be shown that ΠP and NP always satisfy the inclusion relation $\Pi P \supseteq NP$, provided that $\pi_{A(x)}$ is normalized, i.e.,

$$\forall x, \mu_{NP}(x) \leq \mu_{\Pi P}(x).$$

Thus, in case of incomplete information, we are able to compute the set of items which more or less *possibly* satisfy an elementary condition and to distinguish among them the items which more or less *certainly* satisfy this condition. In case of non-fuzzy requests (i.e., P is an ordinary subset of \mathcal{D}), a stronger inclusion holds since then NP is included in the core of ΠP . When the information is precise, i.e., $\pi_{A(x)}$ is equal to 1 for one element d and is zero elsewhere in $\mathcal{D} \cup \{e\}$, it can be checked that $\mu_{\Pi P}(x) = \mu_P(A(x)) = \mu_{NP}(x)$.

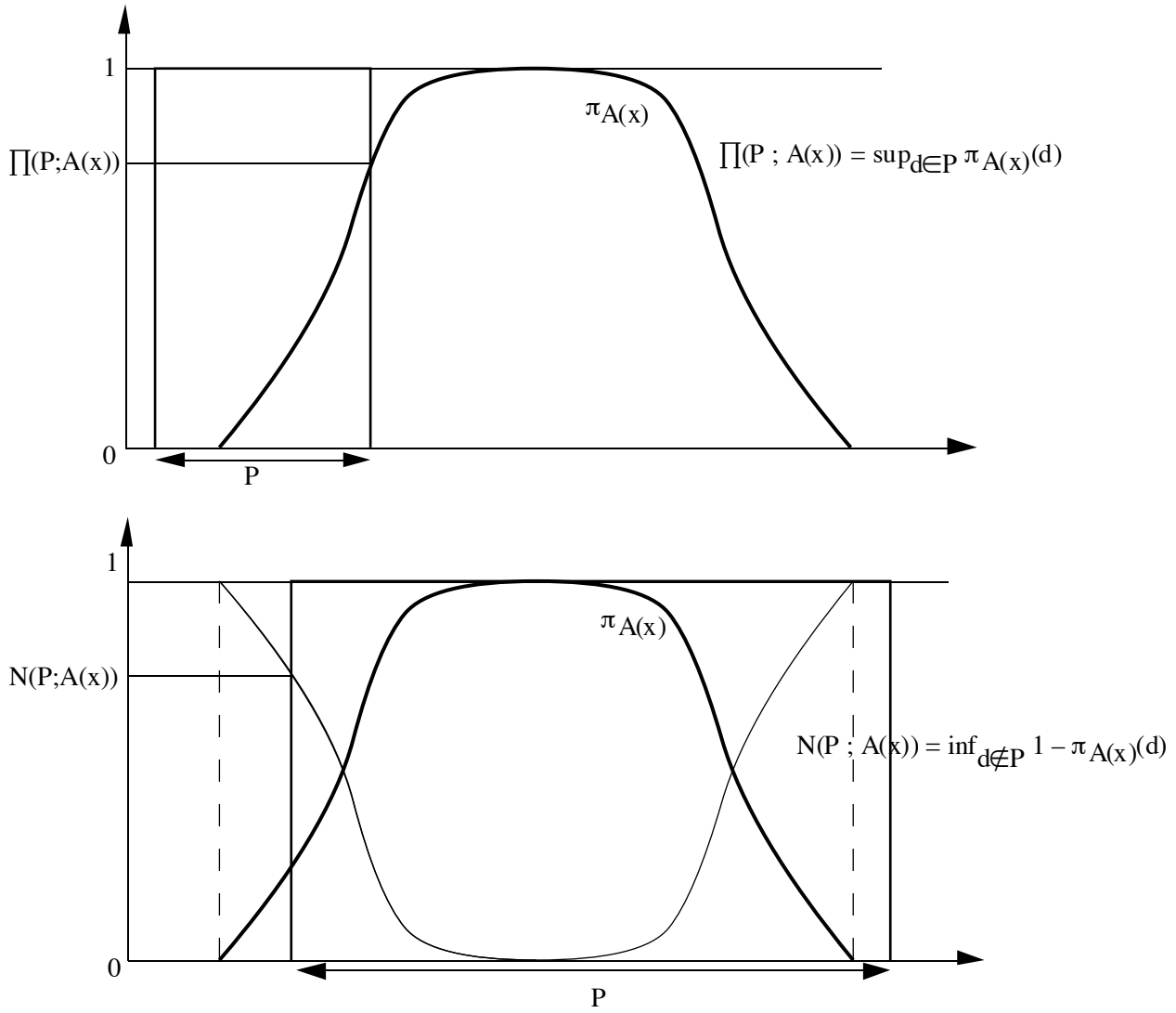


Figure 4 : Computation of possibility and necessity degrees (non-fuzzy query)

N.B. : Similar pattern matching degrees can be defined in the belief function framework where P and $A(x)$ are represented in terms of basic probability assignments (Dubois, Prade and Testemale, 1986).

Selections involving disjunction, conjunction or negation of elementary conditions can be handled using the following basic relations of possibility theory, which express decomposability properties of possibility and necessity degrees with respect to conjunctions and disjunctions provided that the attribute values are logically independent (for any value of $A_1(x)$, all values compatible with $\pi_{A_2(x)}$ are allowed and conversely) :

$$\begin{aligned}
 N(P ; A(x)) &= 1 - \Pi(\bar{P} ; A(x)) \\
 N(P_1 \times P_2 ; A_1(x) \times A_2(x)) &= \min(N(P_1 ; A_1(x)), N(P_2 ; A_2(x))) \\
 \Pi(P_1 + P_2 ; A_1(x) \times A_2(x)) &= \max(\Pi(P_1 ; A_1(x)), \Pi(P_2 ; A_2(x))) \\
 N(P_1 + P_2 ; A_1(x) \times A_2(x)) &= \max(N(P_1 ; A_1(x)), N(P_2 ; A_2(x))) \\
 \Pi(P_1 \times P_2 ; A_1(x) \times A_2(x)) &= \min(\Pi(P_1 ; A_1(x)), \Pi(P_2 ; A_2(x)))
 \end{aligned}$$

where

- the attribute A_i and the subset P_i ($i = 1, 2$) refer to the same domain,
- the overbar denotes the set complementation (defined by $\mu_{\bar{P}}(d) = 1 - \mu_P(d)$),

- $A_1(x) \times A_2(x)$ denotes an extended Cartesian product, expressing a conjunction, defined by

$$\pi_{A_1(x) \times A_2(x)}(d_1, d_2) = \min(\pi_{A_1(x)}(d_1), \pi_{A_2(x)}(d_2))$$

$P_1 \times P_2$ is similarly defined,

- $P_1 + P_2 = \overline{\overline{P_1} \times \overline{P_2}}$ expresses a disjunction, namely

$$\mu_{P_1+P_2}(d_1, d_2) = \max(\mu_{P_1}(d_1), \mu_{P_2}(d_2)).$$

Note that the above expressions of $N(P_1 + P_2 ; A_1(x) \times A_2(x))$ and $\prod(P_1 \times P_2 ; A_1(x) \times A_2(x))$ require the logical independence of the attribute values respectively restricted by $\pi_{A_1(x)}$ and $\pi_{A_2(x)}$, to be valid. See (Dubois, Prade and Testemale, 1988) for weighted conjunctions and disjunctions. These combination formulae are consistent with the fuzzy set operations (based on min and max), when the available information becomes precise, since then the measures of possibility and necessity become equal to a membership degree.

Selections involving fuzzy comparators, e.g., approximate equalities, strong inequalities, can be also easily handled in this framework ; see Prade and Testemale (1984).

The case of other fuzzy set combination operations in compound requests (e.g., product, arithmetic mean), for which no decomposition formula exists for the possibility and necessity measures in presence of incomplete information, can be dealt with by using a fuzzy-real-valued compatibility degree for estimating the agreement between the information and what is required. Then an extended version of the considered combination operation is performed on these fuzzy real values and finally a possibility and a necessity degree can be extracted in a standard way from the global compatibility measure which has been thus computed. The reader is referred to Chapter 3, pp. 98-99 and Chapter 4, pp. 125-126 of (Dubois and Prade, 1988), for detailed definitions and justifications.

For simplicity, we have focused on the selection operation. Queries demanding an extended join operation on relational tables containing fuzzy information are discussed in (Prade and Testemale, 1984).

Possibility theory offers a powerful tool for the representation and the treatment of flexible queries as well as partial information. In spite of the apparent complexity of the expressions of the possibility and necessity degrees, the approach is computationally tractable at least when we restrict ourselves to possibility distributions which are defined on small-sized discrete domains or whose shape is trapezoidal when the domain is a continuum. See Dubois, Mo and Prade (1992) for the use of fuzzy discrimination trees in the pattern matching procedure. Moreover, extensions of indexation techniques have been proposed (Bosc and Galibourg, 1989).

The approach is robust due to the use of the operations max and min, which are not very sensitive to small variations. In practice, it is sufficient to elicitate possibility distributions in a rough way ; i.e., identifying what values are completely impossible, what are the values which are the most possible ones, and then remembering that it is mainly the ordering of possibility degrees with is meaningful in possibility theory, as already explained in 4.1.3.

6. Integrity Constraints and Fuzzy Functional Dependencies

Integrity constraints have been recognized as essential in database management systems since they provide users with the automatic handling of data properties which should be maintained at any time. However, some people have pointed out the fact that sometimes exceptions to these properties occur and should be accepted since they represent actual (although exceptional) situations in the real world. Soft integrity constraints can appear as a rather natural answer to deal with exceptions (for instance, "generally a manager earns more than his subordinates"). Moreover, it is clear that many

data properties (not necessarily constraints) can only be viewed in terms of soft conditions. This field has not received a lot of attention except functional dependencies for which several soft extensions have been proposed.

Soft constraints (expressing either qualitative relationships, like "the larger X, the smaller Y", or allowing for tolerance, exceptions, e.g. "salaries of all employees are approximately equal") may have different uses and associated interests : i) integrity checking is clearly the most important use for constraints and we believe that many practical constraints are in fact fuzzy, ii) providing synthetic (or cooperative) answers and iii) computing default values for missing data. This last use is investigated with a fuzzy approach in (Arrazola et al., 1989).

In the usual framework, functional dependencies have been widely studied since they capture some kind of redundancy and can be used for the design of "good" relational schemas, i.e., which have a robust behaviour when updates are made (basically because the handling of dependencies reduces to the checking of the unicity of key values).

Classical functional dependencies are expressed as

$X \rightarrow Y$ is valid on a relation R iff for any pair of tuples t_1, t_2 of R : $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$.
(where $t[X]$ denotes the projection of the tuple t on the set of attributes X).

In the fuzzy framework, one can think of extending the above expression in different ways : i) replacement of the equality by a similarity relation μ_{EQ_i} ; ii) choice of an implication operator relating degree of satisfaction or of uncertainty associated with (extended) equalities ; iii) weakening of the universal quantification ($\forall t_1, t_2 \in R$) by "for most of the pairs" Depending on the modelling of the implication in the fuzzy case, various kinds of dependencies can be expressed. It may be dependencies stating that the more similar $t_1[X]$ and $t_2[X]$, the more *certain* the identity (or the similarity) of $t_1[Y]$ and $t_2[Y]$, or purely gradual dependencies expressing that the more similar $t_1[X]$ and $t_2[X]$, the more similar $t_1[Y]$ and $t_2[Y]$. See (Dubois and Prade, 1991). These generalized dependencies may include classical functional dependency as a particular case. It will be so if the fuzzy equality relations are such that $\mu_{EQ}(a,b) = 1$ if and only if $a = b$. Then they are stronger since they require equalities in the classical cases and approximate equalities in other cases. Otherwise they are not comparable. People have mainly dealt with the first two points (Anvari and Rose, 1987 ; Raju and Majumdar, 1988 ; Liu, 1992) and have paid attention to the properties of fuzzy functional dependencies (reflexivity, augmentation, transitivity,...). In so doing, it has been shown that Armstrong's axioms were still defining a valid and complete system for the deduction of all dependencies given a starting set when appropriate choices were made for the implication.

Several approaches to the definition of extended functional dependencies have been proposed in the literature and we give three of them. In (Raju and Majumdar, 1988), the authors advocate for choosing the definition:

$X \rightarrow Y$ is valid on R iff $\mu_{EQ_1}(t_1[X] = t_2[X]) \leq \mu_{EQ_2}(t_1[Y] = t_2[Y])$ where EQ_i is a similarity function such that $\mu_{EQ_i}(a,a) = 1$, defined on the domain of X if $i = 1$ and Y if $i = 2$.

In (Chen et al., 1991), the authors suggest the choice of another implication (Gödel's one) :

$X \rightarrow_{\alpha} Y$ is valid on R iff $\min_{t_1, t_2 \in R} I(\mu_{EQ_1}(t_1[X] = t_2[X]), \mu_{EQ_2}(t_1[Y] = t_2[Y])) \geq \alpha$

with $I(a,b) = 1$ if $a \leq b$, $I(a,b) = b$ otherwise.

However, with this definition, a dependency may fail because of following not very similar in X are still less similar in Y. That is why, in (Cubero and Vila, 1992), the following definition is proposed

$$X \rightarrow_{\alpha, \beta} Y \text{ iff } \forall t_1, t_2 \in R : \mu_{EQ_1}(t_1[X] = t_2[X]) \geq \alpha \Rightarrow \mu_{EQ_2}(t_1[Y] = t_2[Y]) \geq \beta.$$

The issue of replacing a relation by two of its projections, when a fuzzy functional dependency holds, has been addressed in (Raju and Majumdar, 1988) and the only condition is that the similarity relations used are such that $\mu_{EQ_1}(a, b) = 1$ iff $a = b$. However, it should be noted that fuzzy functional dependency checking cannot reduce to uniqueness of keys inside relations. Consequently, these dependencies cannot affect the design process of a database.

When data are precisely known the semantics of fuzzy functional dependencies in the above-mentioned cases is rather clear. However, when data are represented by possibility distributions, the calculus of similarity between two distributions becomes crucial and the fact that two distributions are identical does not provide any information about the similarity of the actual values themselves. Consequently, the use of this notion should be performed carefully and it is mandatory to state clearly the meaning of a dependency in case of ill-known values.

8. Concluding Remarks

Before concluding let us briefly mention other uncertainty management problems in data bases where fuzzy set and possibility theory-based methods might be useful :

- data base updates : this problem has been particularly investigated in logical deductive data bases (Winslett, 1990 ; Katsuno and Mendelzon, 1991). In possibility theory, so-called possibilistic imaging (closely related to the fuzzy set extension principle) (Dubois and Prade, 1992) has been proved in perfect agreement with Katsuno and Mendelzon (1991)'s postulates for updates. Generally speaking possibility theory enables us to express preferences among candidates for updating a value and to express the imprecision and the uncertainty pervading laws of evolution of the real world.
- fuzziness and uncertainty in object-oriented representations : the reader is referred to (Dubois, Prade and Rossazza, 1991) for a preliminary study, where an object-centered representation is presented, where both a range of allowed values and a range of typical values can be specified for the attributes describing a class. These ranges may be fuzzy. Then various kinds of (graded) inclusion relations can be defined between classes. Another approach is presented in (Van Gyseghem, De Caluwe and Vandenberghe, 1993) ; see also Mouaddib (1992).
- linguistic summaries : Yager (1982) has advocated the use of fuzzy sets and fuzzy quantifiers in linguistic summaries of the form "Most A's are B". Other summaries expressing gradual relationship such as "the older a person, the bigger the salary" would be also worth-producing. More generally, fuzzy sets may be useful for expressing typical values of attributes for classes of items, or outstanding exceptions to rules.

Imprecision and uncertainty are almost unavoidable when we want to represent and to store the available information as it is, without losing a significative part of it. Allowing for flexible queries seems also very desirable. Fuzzy sets and possibility theory offers a technically sound and powerful framework for dealing both with imprecision and uncertainty and for handling flexible queries. Clearly fuzzy set methods are good for modelling preference among acceptable values in a simple way when expressing queries. Even if only prototypes of fuzzy versions of query languages like SQL have been developed in laboratories until now, due to the smallness of the research community in fuzzy data bases, more applications can be expected in the future. Already the interest in fuzzy information retrieval systems has considerably increased in the last past years in Japan especially. The handling of imprecise, uncertain, vague information in data bases systems is certainly also of practical interest, at least to go a bit further than the treatment of standard null values. Among application-oriented works using the fuzzy pattern matching techniques (presented in Section 5.2) in information systems, let us mention the systems CLASSIC and FLORAN respectively developed by Granger (1988) and Salotti (1993). The implementations which already exist (especially for fuzzy querying

systems) tend to show that, when clearly identified, problems can be solved in a reasonable computational time.

Acknowledgements

The paper has profited from many remarks made on previous drafts by Robert Demolombe, Guy Hulin and Philippe Smets.

References

- Anvari M., Rose G.F. (1987) Fuzzy relational database. In : Analysis of Fuzzy Information — Vol. II : Artificial Intelligence and Decision Systems (J.C. Bezdek, ed.), CRC Press, Boca Raton, FL., 203-212.
- Arrazola I., Plainfossé A., Prade H., Testemale C. (1989) Extrapolation of fuzzy values from incomplete data bases. Information Systems, 14(6), 487-492.
- Baldwin J.F., Zhou S.Q. (1984) A fuzzy relational inference language. Fuzzy Sets and Systems, 14, 155-174.
- Barbará D., Garcia-Molina H., Porter D. (1992) The management of probabilistic data. IEEE Trans. on Knowledge and Data Engineering.
- Biskup J. (1979) A formal approach to null values in database relations. Proc. of the Workshop on Formal Bases for Data Bases (H. Gallaire, J.M. Nicolas, eds.), Toulouse, Dec. 12-14, 1979, Plenum Press, 1980.
- Biswas G., Bezdek J.C., Marques M., Subramanian V. (1987) Knowledge-assisted document retrieval — Part I : The natural-language interface ; Part II : The retrieval process. J. of the American Society for Information Science, Part I : 38(2), 83-96 ; Part II : 38(2), 97-110.
- Bollmann P., Konrad E. (1976) Fuzzy document retrieval. Proc. of the 3rd Europ. Meeting on Cybernetics and Systems Research, Vienna, 355-363.
- Bookstein A. (1980) Fuzzy requests : an approach to weighted Boolean searchers. J. Am. Soc. Inform. Sci., 31, 240-247.
- Bordogna G., Carrara P., Pasi G. (1991) Query term weights as constraints in fuzzy information retrieval. Information Processing and Management, 27(1), 15-26.
- Bosc P., Galibourg M. (1989) Indexing principles for a fuzzy data base. Information Systems, 14(6), 493-499.
- Bosc P., Galibourg M., Hamon G. (1988) Fuzzy querying with SQL : extensions and implementation aspects. Fuzzy Sets and Systems, 28, 333-349.
- Bosc P., Pivert O. (1991) About equivalences in SQL_f, a relational language supporting imprecise querying. Proc. Inter. Fuzzy Engineering Symp. (IFES'91), Yokohama, Japon, Nov. 13-15, 309-320.
- Bosc P., Pivert O. (1992a) Some approaches for relational databases flexible querying. Int. J. of Intelligent Information Systems, 1, 323-354.
- Bosc P., Pivert O. (1992b) Discriminated answers and databases : fuzzy sets as a unified expression means. Proc. 1st Inter. IEEE Conf. on Fuzzy Systems (FUZZ-IEEE'92), San Diego, CA, March, 745-752.
- Bosc P., Pivert O. (1992c) On the evaluation of fuzzy quantified queries in a database management system. Proc. of the North American Fuzzy Logic Processing Society Conf. (NAFIPS'92), Puerto Vallarta, Mexico, December 15-17.

- Bosc P., Pivert O. (1992d) Some properties of alpha-cuts of fuzzy predicates. Proc. of the 11th Meeting on Cybernetics and Systems Research, Vienne, Austria.
- Buckles B.P., Petry F.E. (1982) A fuzzy representation of data for relational databases. Fuzzy Sets and Systems, 5, 213-226.
- Buckles B.P., Petry F.E. (1987) Generalized database and information systems. In : Analysis of Fuzzy Information — Vol. II : Artificial Intelligence and Decision Systems (J.C. Bezdek, ed.), CRC Press, Boca Raton, FL., 177-201.
- Cayrol M., Farreny H., Prade H. (1982) Fuzzy pattern matching. Kybernetes, 11, 103-116.
- Chang C.L. (1982) Decision support in an imperfect world. Tech. Rep. RJ3421 (40687), IBM Research Lab., Computer Science, San Jose, CA.
- Chen G.Q., Vandenbulcke J. (1999) A step towards the theory of fuzzy relational database design. Proc. of the Inter. Fuzzy Systems Assoc. (IFSA'91) Congress, Brussels, Belgium, Vol. : Computer, Management & Systems Science (R. Lowen, M. Roubens, eds.), 44-47.
- Cubero J.C., Vila M.A. (1992) A new definition of fuzzy functional dependency in fuzzy relational databases. Proc. of the 4th Inter. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'92) (B. Bouchon-Meunier, L. Valverde, R.R. Yager, eds.), Palma de Mallorca, July 6-10, 1992, Lecture Notes in Computer Science, Springer Verlag, Berlin, 239-242.
- Demolombe R. (1992) Uncertainty in intelligent data bases. 1st Inter. Seminar UMIS.
- Dubois D., Lang J., Prade H. (1989) Automated reasoning using possibilistic logic : semantics, belief revision and variable certainty weights. Preprints of the 5th Workshop on Uncertainty in Artificial Intelligence, Windsor, Ontario, Aug. 18-20, 1989, 81-87. Revised version in IEEE Trans. on Data and Knowledge Engineering, 1993, to appear.
- Dubois D., Lang J., Prade H. (1991) Handling uncertainty, context, vague predicates, and partial inconsistency in possibilistic logic. Preprints of the Fuzzy Logic in Artificial Intelligence Workshop held in conjunction with IJCAI'91, Sydney, Australia, Aug. 25, 13-23.
- Dubois D., Lang J., Prade H. (1992) Dealing with multi-source information in possibilistic logic. Proc. of the 10th Europ. Conf. on Artificial Intelligence (ECAI'92) (B. Neumann, ed.), Vienna, Austria, Aug. 3-7, 38-42.
- Dubois D., Lang J., Prade H. (1993) Possibilistic logic. In : Handbook of Logic in Artificial Intelligence and Logic Programming, Vol. 3 (D.M. Gabbay, ed.), Oxford University Press, to appear.
- Dubois D., Mo X., Prade H. (1991) Fuzzy discrimination trees. Proc. of the Inter. Fuzzy Engineering Symp. (IFES'91), Yokohama, Japan, Nov. 13-15, Vol. 1 : Fuzzy Engineering toward Human Friendly Systems, 250-260.
- Dubois D., Prade H. (1985) A review of fuzzy set aggregation connectives. Information Sciences, 36, 85-121.
- Dubois D., Prade H. (1986) Weighted minimum and maximum operations in fuzzy set theory. Information Sciences, 39, 205-210.
- Dubois D., Prade H. (with the collaboration of Farreny H., Martin-Clouaire R., Testemale C.) (1988a) Possibility Theory : an Approach to Computerized Processing of Uncertainty. Plenum Press, New York.
- Dubois D., Prade H. (1988b) Incomplete conjunctive information. Comput. Math. Appl., 15(10), 797-810.
- Dubois D., Prade H. (1990) Measuring properties of fuzzy sets : a general technique and its use in fuzzy query evaluation. Fuzzy Sets and Systems, 38, 137-152.
- Dubois D., Prade H. (1991) Certainty and uncertainty of (vague) knowledge and generalized dependencies in fuzzy data bases. Proc. of the Inter. Fuzzy Engineering Symp. (IFES'91), Yokohama, Japan, Nov. 13-15, 239-249.

- Dubois D., Prade H. (1992) Belief revision and updates in numerical formalisms — An overview, with new results for the possibilistic framework —. Technical Report, I.R.I.T., Université Paul Sabatier, Toulouse, France.
- Dubois D., Prade H., Rossazza J.P. (1991) Vagueness, typicality and uncertainty in class hierarchies. Int. J. of Intelligent Systems, 6, 167-183.
- Dubois D., Prade H., Testemale C. (1986) Fuzzy pattern matching with extended capabilities : proximity notions, importance assessment, random sets. Proc. of the North American Fuzzy Information Processing Society (NAFIPS'86) : "Recent Developments in the Theory and Applications of Fuzzy Sets" (W. Bandler, A. Kandel, eds.), New Orleans, LA, June 2-4, 125-139.
- Dubois D., Prade H., Testemale C. (1988) Weighted fuzzy pattern matching. Fuzzy Sets and Systems, 28(3), 313-331.
- Friedman J.H., Baskett F., Shustek L.J. (1975) An algorithm for finding nearest neighbors. IEEE Trans. on Computers, 1001-1006.
- Granger C. (1988) An application of possibility theory to object recognition. Fuzzy Sets and Systems, 28, 351-362.
- Ichikawa T., Hirakawa M. (1986) ARES : a relational database with the capability of performing flexible interpretation of queries. IEEE Trans. on Software Engineering, 12(5), 624-634.
- Kacprzyk J., Zadrozny S., Ziolkowski A. (1989) Fquery III+ : a "human-consistent" database querying system based on fuzzy logic with linguistic quantifiers. Information Systems, 14(6), 443-453.
- Katsuno H., Mendelzon A.O. (1991) On the difference between updating a knowledge base and revising it. Proc. of the 2nd Inter. Conf. on Principles of Knowledge Representation and Reasoning (KR'91) (J. Allen, R. Fikes, E. Sandewall, eds.), Cambridge, Mass., April 22-25, 387-394.
- Kohout L.J., Keravnou E., Bandler W. (1984) Automatic documentary information retrieval by means of fuzzy relational products. In : Fuzzy Sets and Decision Analysis (H.J. Zimmermann, L.A. Zadeh, and B.R. Gaines, eds.), TIMS/Studies in the Management Sciences, Vol. 20, North-Holland, Amsterdam, 383-404.
- Kraft D.H., Buell D.A. (1983) Fuzzy sets and generalized Boolean retrieval systems. Int. J. of Man-Machine Studies, 19, 45-56.
- Kunii T.L. (1976) Dataplan : an interface generator for database semantics. Information Sciences, 10, 279-298.
- Lacroix M., Lavency P. (1987) Preferences : putting more knowledge into queries. Proc. of the 13th Very Large Data Bases Conference, Brighton, 217-225.
- Li D.Y., Liu D.B. (1990) A Fuzzy PROLOG Database System. Research Studies Press Ltd., Taunton, Somerset.
- Liu W.Y. (1992) The reduction of the fuzzy data domain and fuzzy consistent join. Fuzzy Sets and Systems, 50, 89-96.
- McClelland E., Trueblood R., Eastman C. (1988) Two approximate operators for a data base query language : sounds_like and close_to. IEEE Trans. on Systems, Man and Cybernetics, 18(6), 873-884.
- Mendelson H., Saharia A.N. (1986) Incomplete information costs and database design. ACM Trans. on Database Systems, 11, 159-185.
- Miyamoto S. (1990) Fuzzy Sets in Information Retrieval and Cluster Analysis. Kluwer Academic Publ., Dordrecht.
- Motro A. (1988) VAGUE : a user interface to relational databases that permits vague queries. ACM Trans. on Office Information Systems, 6, 187-214.

- Mouaddib N. (1992) The nuanced identification in a database — The nuanced relational division. Proc. of the Inter. Conf. on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'92), Mallorca, Spain, July 6-10, 267-270.
- Murai T., Miyakoshi M., Shimbo M. (1989) A fuzzy document retrieval method based on two-valued indexing. Fuzzy Sets and Systems, 30, 103-120.
- Negoita C.V., Flondor P. (1976) On fuzziness in information retrieval. Int. J. of Man-Machine Studies, 8, 711-716.
- Nomoto K., Wakayama S., Kirimoto T., Ohashi Y., Kondo M. (1990) A document retrieval system based on citations using fuzzy graphs. Fuzzy Sets and Systems, 38, 207-222.
- Ogawa Y., Morita T., Kobayashi K. (1991) A fuzzy document retrieval system using the keyword connection matrix and a learning method. Fuzzy Sets and Systems, 39, 163-179.
- Pawlak Z. (1991) Rough Sets — Theoretical Aspects of Reasoning about Data. Kluwer, Dordrecht.
- Potoczny H.B. (1984) On similarity relations in fuzzy relational databases. Fuzzy Sets and Systems, 12, 231-235.
- Prade H. (1982) The connection between Lipski's approach to incomplete information data bases and Zadeh's possibility theory. In : Systems Science and Science (Proc. of the 16th Annual Meeting of the Society for General Systems Research with the American Association for the advancement of Science, Washington, D.C., Jan. 5-9, 1992) (L. Troncale, ed.), 402-408.
- Prade H. (1984) Lipski's approach to incomplete information data bases restated and generalized in the setting of Zadeh's possibility theory. Information Systems, 9(1), 27-42.
- Prade H. (1990) A two-layer fuzzy pattern matching procedure for the evaluation of conditions involving vague quantifiers. J. of Intelligent Robotic Systems, 3, 93-101.
- Prade H., Testemale C. (1984) Generalizing database relational algebra for the treatment of incomplete/uncertain information and vague queries. Information Sciences, 34, 115-143.
- Prade H., Testemale C. (1987a) Representation of soft constraints and fuzzy attribute values by means of possibility distributions in databases. In : Analysis of Fuzzy Information — Vol. 2 : Artificial Intelligence and Decision Systems (J.C. Bezdek, ed.), CRC Press, Boca Raton, Fl., 213-229.
- Prade H., Testemale C. (1987b) Fuzzy relational databases : representational issues and reduction using similarity measures. J. of the American Society for Information Science, 38(2), 118-126.
- Prade H., Testemale C. (1987c) Application of possibility and necessity measures to documentary information retrieval. In : Uncertainty in Knowledge-Based Systems (B. Bouchon, and R.R. Yager, eds.), Lecture Notes in Computer Science, Vol. 286, Springer Verlag, Berlin, 265-274.
- Rabitti F. (1990) Retrieval of multimedia documents by imprecise query specification. Lecture Notes on Computer Science, Vol. 416, 203-218.
- Radecki T. (1983) Generalized Boolean methods of information retrieval. Int. J. of Man-Machine Studies, 18(5), 407-439.
- Raju K.V.S.V.N., Majumdar A.K. (1988) Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. ACM Trans. on Database Systems, 13(2), 129-166.
- Reisinger L. (1974) On fuzzy thesauri. Proc. of the Symp. on Comput. Stat. (Compstat'74) (G. Bruckman, F. Ferschl, L. Schmetterer, eds.), Physica-Verlag, Vienna, 119-127.
- Rundensteiner E.A., Hawkes L.W., Bandler W. (1989) On nearness measures in fuzzy relational data models. Int. J. of Approximate Reasoning, 3, 267-298.
- Ruspini E.H. (1987) Issues in the representation of imprecision and uncertainty in information systems. In : Analysis of Fuzzy Information — Vol. II : Artificial Intelligence and Decision Systems (J.C. Bezdek, ed.), CRC Press, Boca Raton, Fl., 231-239.

- Salotti S. (1993) Fuzzy pattern matching and object-centered representation for case-based reasoning: the FLORAN system. To appear.
- Sanchez E. (1989) Importance in knowledge systems. Information Systems, 14(6), 455-464.
- Shenoi S., Melton A., Fan L.T. (1992) Functional dependencies and normal forms in the fuzzy relational database model. Information Sciences, 60, 1-28.
- Steger N., Schmidt H., Guntzer U., Kiessling W. (1989) Semantics and efficient compilation for quantitative deductive databases. Proc. of the 5th Inter. Conf. on Data Engineering, Los Angeles, CA, Feb. 6-10, 660-668.
- Su S.Y.W., Dujmovic J., Batory D.S., Navathe S.B., Elnicki R. (1987) A cost-benefit decision model: Analysis, comparison, and selection of data management systems. ACM Trans. on Database Systems, 12(3), 472-520.
- Tahani V. (1977) A conceptual framework for fuzzy query processing — A step toward very intelligent database systems. Information Processing Management, 13, 289-303.
- Testemale C. (1990) Traitement d'informations incomplètes dans les bases de données : des valeurs nulles aux informations floues. In : Modèles Logiques et Systèmes d'Intelligence Artificielle (L. Iturrioz, A. Dussauchoy, eds.), Hermès, Paris, 291-312.
- Umano M. (1982) FREEDOM-0 : a fuzzy database system. In : Fuzzy Information and Decision Processes (M.M. Gupta, E. Sanchez, eds.), North-Holland, Amsterdam, 339-347.
- Umano M. (1983) Retrieval from fuzzy database by fuzzy relational algebra. In : Fuzzy Information, Knowledge Representation and Decision Analysis (E. Sanchez, M.M. Gupta, eds.), Pergamon Press, New York, 1-6.
- Umano M., Fukami S. (1991) Perspectives of fuzzy databases. Japanese J. of Fuzzy Theory and Systems (English translation, Allerton Press, New York), 3, 75-91.
- Vandenberghe R., Van Schooten A., De Caluwe R., Kerre E.E. (1989) Some practical aspects of fuzzy database techniques : an example. Information Systems, 14(6), 465-472.
- Van Gyseghem N., De Caluwe R., Vandenberghe R. (1993) UFO : uncertainty and fuzzyness in an object-oriented model. Proc. 2nd Inter. IEEE Conf. on Fuzzy Systems (FUZZ-IEEE'93), San Francisco, CA, March, 773-778.
- Winslett M. (1990) Updating Logical Databases. Cambridge University Press, Cambridge, UK.
- Wong E.A. (1982) Statistical approach to incomplete information in database systems. ACM Trans. on Database Systems, 7(3), 470-488.
- Yager R.R. (1980) A logical on-line bibliographic searcher : an application of fuzzy sets. IEEE Trans. on Systems, Man and Cybernetics 10(1), 51-53.
- Yager R.R. (1982) A new approach to the summarization of data. Information Sciences, 28, 69-86.
- Yager R.R. (1987) A note on weighted queries in information retrieval systems. J. of the American Society of Information Sciences, 38, 23-24.
- Yager R.R. (1988) On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Trans. on Systems, Man and Cybernetics, 18, 183-190.
- Yager R.R. (1991a) Connectives and quantifiers in fuzzy sets. Fuzzy Sets and Systems, 40, 39-76.
- Yager R.R. (1991b) Fuzzy quotient operators for fuzzy relational databases. Proc. of the Inter. Fuzzy Engineering Symp. (IFES'91), Yokohama, Japan, Nov. 13-15, 289-296.
- Zadeh L.A. (1965) Fuzzy sets. In : Information and Control, Vol. 8, Academic Press, New York, pp. 338-353.
- Zadeh L.A. (1978) Fuzzy sets as a basis for a theory of possibility. Fuzzy Sets and Systems, 1, pp. 3-28.
- Zadeh L.A. (1983) A computational approach to fuzzy quantifiers in natural language. Computer and Mathematics with Applications, 9(1), 149-184.

- Zemankova M. (1989) FILIP: a fuzzy intelligent information system with learning capabilities. Information Systems, 14(6), 473-486.
- Zemankova-Leech M., Kandel A. (1984) Fuzzy Relation Data Bases — A Key to Expert Systems. Verlag TÜV Rheinland, Köln (180 pages).
- Zenner R.B.R.C., de Caluwe R.M.M., Kerre E.E. (1985) A new approach to information retrieval systems using fuzzy expressions. Fuzzy Sets and Systems, 17, 9-22.
- Zimanyi E. (1992) Incomplete and uncertain information in relational databases. Thèse de Docteur, Université Libre de Bruxelles, Belgique.
- Zimmermann H.J., Zysno P. (1980) Latent connectives in human decision making. Fuzzy Sets and Systems, 4, 37-51.
- Zvieli A., Chen P.P. (1986) Entity-relationship modeling and fuzzy databases. Proc. of the 2nd IEEE Data Engineering Conf., Los Angeles, Feb. 5-7, 320-327.