Fuzzy Sets Defined on a Hierarchical Domain

Rallou Thomopoulos, Patrice Buche, Ollivier Haemmerlé

Fuzzy Sets Defined on a Hierarchical Domain

Rallou Thomopoulos[1,2], Patrice Buche[3], and Ollivier Haemmerlé[4]

***Abstract*— This paper presents a new type of fuzzy sets, called "Hierarchical Fuzzy Sets", that apply when the considered domain of values is not "flat", but contains values that are more specific than others according to the "kind of" relation. We study the properties of such fuzzy sets, that can be defined in a short way on a part of the hierarchy, or exhaustively (by their "closure") on the whole hierarchy. We show that hierarchical fuzzy sets form equivalence classes in regard to their closures and that each class has a particular representative called "minimal fuzzy set". We propose a use of this minimal fuzzy set for query enlargement purposes and thus present a methodology for hierarchical fuzzy set generalization. We finally present an experimental evaluation of the theoretical results described in the paper, in a practical application.**

***Index Terms*— Fuzzy set; Uncertainty, "fuzzy" and probabilistic reasoning; Object hierarchies; Relaxation; Knowledge retrieval.**

I. Introduction

In classic querying systems, the queries sent by the users are all-or-nothing queries: a value belongs to the users' selection criteria or does not. In soft querying [1], the users have the possibility to express preferences in their selection criteria. In this context, fuzzy sets, which are more generally used to represent concepts whose borders are not strictly delimited, can be used to define flexible selection criteria, by associating a preference degree with every candidate value. As a parallel issue, classic databases contain precise data, which are not expected to be ill-known. In possibilistic databases [2], an ill-known datum is represented by a possibility distribution, which associates a possibility degree with every candidate value (with the hypothesis that only one of these values is the effective one).

[1] INRA, IATE Joint Research Uunit (bât. 31), 2 place P. Viala, F-34060 Montpellier Cedex 1, email: rallou@ensam.inra.fr

[2] Associate Researcher of LIRMM (CNRS & Université Montpellier II), 161 rue Ada, F-34392 Montpellier Cedex 5

[3] INRA, Mét@risk Research Uunit, 16 rue Claude Bernard, F-75231 Paris Cedex 5, email: patrice.buche@inapg.inra.fr

[4] GRIMM-ISYCOM, Université Toulouse le Mirail, Département Mathématiques-Informatique, 5 allées Antonio Machado, F-31058 Toulouse Cedex, email: ollivier.haemmerle@univ-tlse2.fr

These two approaches, fuzzy sets [3] and possibility theory [4], constitute a homogeneous formalism in two different uses. In both uses, an order relation is defined on a domain of values. In this paper, we consider the case when the candidate values of a selection criterion in the first use, or of an ill-known datum in the second use, are not "flat" domain values but are elements of a hierarchy, partially ordered by the "kind of" relation: some of the values are more specific than others.

Contrary to a fuzzy set defined on a "flat" domain of values, in our case the assumption of independency of the values is not true. Therefore two order relations - the preference/possibility order relation, and the "kind of" partial order relation - must be put in adequacy. Some of the questions we had to answer were: Does the preference or possibility degree associated with a given value have implications on the degrees associated with the other values of the hierarchy, particularly more specific or more general ones ? What would be the meaning of two comparable values (with the meaning of the "kind of" relation) associated with different preference or possibility degrees ? Can the hierarchical structure be used to enlarge the users' queries in case of empty answers, while respecting the preference order defined by the users in their selection criteria ?

Previous approaches close to our work are those regarding similar questions in non-fuzzy contexts. In particular, the propagation of preference or possibility degrees in a hierarchy that we propose is in adequacy with the object model, in which a query on a given class is also addressed to the subclasses of this class. Concerning query enlargement, several works such as [5], [6] use a lattice of concepts to generalize unsolvable queries.

In the bibliography concerning the introduction of fuzzy methods, several issues have been dealt with but are quite distant from our concern. We can note two main categories of papers, especially in recent research:

- the use of linguistic labels in ontologies. In

studies about possibilistic ontologies [7], each term of an ontology is considered as a linguistic label and has an associated fuzzy description. Fuzzy pattern matching between different ontologies is then computed using these fuzzy descriptions. This approach is related to those concerning the introduction of fuzzy attribute values in the object model [8];

- the use of fuzzy relations between the terms of a thesaurus. Studies about fuzzy thesauri have discussed different natures of relations between concepts, where relations are gradual and moderated by degrees. Fuzzy thesauri have been considered for instance in [9], [10]. In this approach, a query composed of a set of terms is enlarged to similar terms thanks to fuzzy pseudo-thesauri. Similarity is based on the co-occurrence frequency of terms in a given set of documents.

However in our context the terms of the hierarchy and the relations between terms are not fuzzy.

The present work was applied in the framework of a French national project dedicated to microbiological risk assessment in foods. The examples given in the paper come from this case study. As a first step of the project, scientific data from predictive microbiology were gathered and a querying system was built in order to explore them. The data have two characteristics:

- they are not abundant enough to answer every query, thus there is a need for preference expression (for instance, the users may ask for milk as a first choice or yoghourt as a second choice) in order to make the querying more flexible, as well as for query enlargement (including other dairy products for example) in case of empty or insufficient answers;
- they include ill-known information. For instance, in some kinds of human diseases, the bacterium *Escherichia coli* is suspected to be responsible, but other bacteria like *Shigella* are not excluded.

The food products, like milk or yoghourt, are part of a hierarchy of substrates, in which, for instance, *Whole milk* is a kind of *Milk*, which is a kind of *Milk product*, etc. In the same way, the bacteria *Escherichia coli* and *Shigella* are part of a hierarchy of micro-organisms.

The methods presented in the paper have been implemented in several representation formalisms: the conceptual graph model, the relational model and XML, and some parts (the definition of hierarchical fuzzy sets more specifically) have been published as they constitute extensions or special uses of these formalisms (see respectively [11] - [12], [13] and [14]). Our goal in this article is to provide a complete theoretical study, including generalization mechanisms – which has never been presented before – apart from the context of a specific data model.

In the following, we firstly remind in Section II the basics of fuzzy sets. In Section III, we develop the notion of hierarchical fuzzy set. In Section IV, we propose a complementary solution to the lack of answers to a query, based on the generalization of a hierarchical fuzzy set. In Section V, we present an experimental evaluation of the proposed methods.

II. Preliminary Notions

In this section, we briefly present fuzzy sets, that will be used in the following to represent the required values in a flexible query or the possible values in an ill-known datum. We also introduce comparisons between fuzzy sets that will be used to compare an ill-known datum to a flexible query.

A. Fuzzy Sets

Fuzzy sets [3] were introduced to represent concepts that are not strictly delimited, like “young” or “far” for instance. Unlike the case of a classic set, an element may belong partially to a fuzzy set.

Definition 1: a **fuzzy set** A on a domain X is defined by a membership function μ_A from X to [0, 1] that associates the degree to which x belongs to A with each element x of X .

The domain X may be continuous or discrete. In this paper, we only deal with discrete domains, as further presented in Section III. Figure 1 illustrates two examples already mentioned above. The fuzzy sets *ProductPreferences* and *ResponsibleBacterium* are also denoted, respectively, *1/Milk + 0.5/Yoghourt*, and *1/Escherichia coli + 0.7/Shigella*, which indicates the degree associated with each element. These fuzzy sets are user-defined, during the choice of the querying selection criteria, or during the entry of an ill-known datum.

We call *support* and *kernel* of a fuzzy set A respectively the sets $support(A) = \{x \in X \mid$

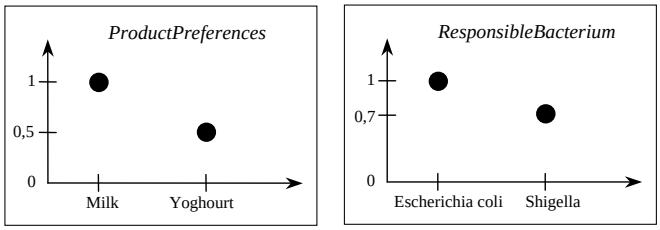


Fig. 1. The fuzzy sets *ProductPreferences* and *ResponsibleBacterium*

$\mu_A(x) > 0\}$ and $kernel(A) = \{x \in X \mid \mu_A(x) = 1\}$.

In the following, we focus on two different comparisons between fuzzy sets: the inclusion relation, that we use to determine in a binary way whether an ill-known datum is an answer to a flexible query or not, and fuzzy pattern matching, which allows to determine in a graduate way whether an ill-known datum somehow answers a flexible query.

B. Comparisons between fuzzy sets

In the most commonly used inclusion relation between fuzzy sets, a fuzzy set A (in our case, an ill-known datum) is included in B (in our case, a flexible query) if its membership function is "below" the membership function of B. More formally:

Definition 2: Let A and B be two fuzzy sets defined on a domain X. A is included in B (denoted $A \subseteq B$) if and only if their membership functions μ_A and μ_B satisfy the condition:

$$\forall x \in X, \mu_A(x) \leq \mu_B(x).$$

Two scalar measures are classically used in fuzzy pattern matching [15] to evaluate the compatibility between an ill-known datum and a flexible query: (i) a possibility degree of matching [4]; (ii) a necessity degree of matching [16].

Definition 3: Let Q and D be two fuzzy sets defined on a domain X and representing respectively a flexible query and an ill-known datum:

- the possibility degree of matching between Q and D, denoted $\Pi(Q; D)$, is an "optimistic" degree of overlapping that measures the maximum compatibility between Q and D, and is defined by
  $\Pi(Q; D) = sup_{x \in X} min(\mu_Q(x), \mu_D(x))$;
- the necessity degree of matching between Q and D, denoted $N(Q; D)$, is a "pessimistic" degree of inclusion that estimates the extent to which it is certain that D is compatible with Q, and is defined by
  $N(Q; D) = inf_{x \in X} max(\mu_Q(x), 1 - \mu_D(x))$.

Although a fuzzy set representing possible values in an ill-known datum and a fuzzy set expressing the user's interests in a query are different concerns, we must note, firstly, that they share the same definition domain (which will be a common hierarchy in the following), and secondly, that their comparisons have been widely studied in the literature [1], [4], [15], [17].

III. Hierarchical fuzzy sets

The notion of hierarchical fuzzy set rose from our need to express fuzzy values in the case where these values are part of taxonomies, as for food products or micro-organisms for example. In our first approach, presented in Section III-A, such a fuzzy set is created directly by the user and defined on a part of the hierarchy. In our second approach, for reasons explained in Section III-B, we extend the fuzzy set to the whole hierarchy, thus obtaining the *closure* of the fuzzy set. Section III-C defines how we extend the comparisons between classic fuzzy sets to hierarchical fuzzy sets. In Section III-D, we show that hierarchical fuzzy sets having the same closure lead to equivalence classes and that each class has one particular representative which is said to be *minimal*.

A. Presentation

The definition domains of the fuzzy sets that we define below are subsets of hierarchies composed of elements partially ordered by the "kind of" relation. An element elt is more general than an element elt' (denoted $elt' \leq elt$), if elt' is a predecessor of elt in the partial order induced by the "kind of" relation of the hierarchy. An example of such a hierarchy is given in Figure 2. A hierarchical fuzzy set is then defined as follows.

Definition 4: A **hierarchical fuzzy set** is a fuzzy set whose definition domain is a subset of the elements of a finite hierarchy partially ordered by the "kind of" relation.

For example, the fuzzy sets *ProductPreferences* and *ResponsibleBacterium* represented in Figure 1 conform to Definition 4. Their definition domains are subsets of the hierarchy given in Figure 2.

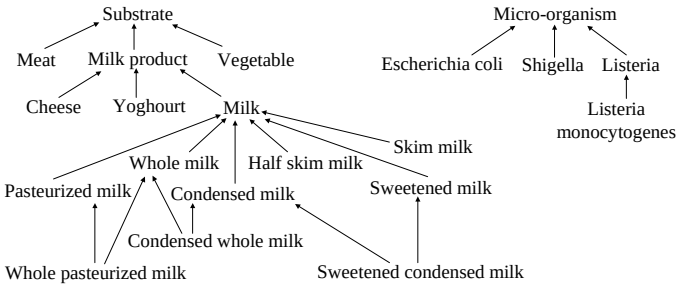


Fig. 2. Example of a hierarchy

We can note that no restriction has been imposed concerning the elements that compose the definition domain of a hierarchical fuzzy set. In particular, the user may associate a given degree d with an element elt and another degree d' with an element elt' more specific than elt. $d' \leq d$ represents a semantic of restriction for elt' compared to elt, whereas $d' \geq d$ represents a semantic of reinforcement for elt' compared to elt.

For example, if there is particular interest in skim milk because the user studies the properties of low fat products, but also wants to retrieve complementary information about other kinds of milk, these preferences can be expressed using for instance the following fuzzy set: *1/Skim milk + 0.5/Milk*. In this example, the element *Skim milk* has a greater degree than the more general element *Milk*, which corresponds to a semantic of reinforcement for *Skim milk* compared to *Milk*. On the contrary, if the user is interested in all kinds of milk, but to a lesser extent in *Condensed milk* because of its smaller water content, the preferences can be expressed using the following fuzzy set: *1/Milk + 0.2/Condensed milk*. In this case, the element *Condensed milk* has a smaller degree than the more general element *Milk*, which corresponds to a semantic of restriction for *Condensed milk* compared to *Milk*.

B. Closure of a hierarchical fuzzy set

We can make two remarks concerning the use of hierarchical fuzzy sets:

- the first one is semantic. Let *1/Skim milk + 0.5/Milk* be an expression of preferences in a query. We can note that this hierarchical fuzzy set implicitly gives information about elements of the hierarchy other than *Skim milk* and *Milk*. For instance, one can deduce that the user does not expect results concerning products like meat or vegetable, even if the degree 0 has not explicitly been associated with these products. One may also assume that any kind of skim milk (sterilized, pasteurized, raw skim milk for example) interests the user with the degree 1;
- the second one is operational. The problem rising from Definition 4 is that two different fuzzy sets on the same hierarchy do not necessarily have the same definition domain, which means they cannot be compared using the classic comparison operations of fuzzy set theory (see Definitions 2, 3). For example, *1/Skim milk + 0.5/Milk* and *1/Milk + 0.2/Condensed milk* are defined on two different subsets of the hierarchy of Figure 2 and thus are not comparable.

These remarks led us to introduce the concept of *closure* of a hierarchical fuzzy set, which is a developed form defined on the whole hierarchy. Intuitively, in the closure of a hierarchical fuzzy set, the "kind of" relation is taken into account by propagating the degree associated with an element to its sub-elements (more specific elements) in the hierarchy. For instance, in a query, if the user is interested in the element *Milk*, we consider that all kinds of *Milk* – *Whole milk*, *Skim milk*, *Pasteurized milk*, etc. – are of interest. On the opposite, we consider that the super-elements (more general elements) of *Milk* in the hierarchy – *Milk product*, *Substrate*, ...– are too general to be relevant for the user's query.

Definition 5: Let F be a hierarchical fuzzy set defined on a subset D of the elements of a hierarchy H. Its membership function is denoted μ_F. The **closure** of F, denoted $clos(F)$, is a hierarchical fuzzy set defined on the whole set of elements of H and its membership function $\mu_{clos(F)}$ is defined as follows.

For each element elt of H, let $E_{elt} = \{elt_1, \ldots, elt_n\}$ be the set of the smallest super-elements of elt in D (in the broad sense, i.e. $elt_i \geq elt$):

- if E_{elt} is not empty, $\mu_{clos(F)}(elt) = max_{1 \leq i \leq n}(\mu_F(elt_i))$;
- otherwise $\mu_{clos(F)}(elt) = 0$.

In other words, the closure of a hierarchical fuzzy set F is built according to the following rules. For each element elt of H:

1) if elt belongs to F, then elt keeps the same

degree in the closure of F (case where $E_{elt} = \{elt\}$);

2) if elt has a unique smallest super-element elt_1 in F, then the degree associated with elt_1 is propagated to elt in the closure of F (case where $E_{elt} = \{elt_1\}$ with $elt_1 > elt$);
3) if elt has several smallest super-elements $\{elt_1, \ldots, elt_n\}$ in F, with different degrees, a choice has to be made concerning the degree that will be associated with elt in the closure. The proposition made in Definition 5 consists in choosing the maximum of the degrees associated with $elt_1, \ldots, elt_n$. This choice is discussed in the following;
4) all the other elements of H, i.e. those that are more general than, or not comparable with the elements of F, are considered as non-relevant. The degree 0 is associated with them (case where $E_{elt} = \emptyset$).

Example 1: Figure 3 shows the closure of the hierarchical fuzzy set *0.8/Milk + 1/Whole milk + 0.3/Condensed milk*.

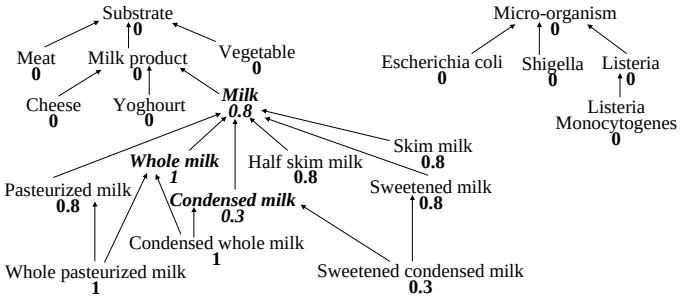


Fig. 3. Closure of a hierarchical fuzzy set

In the hierarchical fuzzy set of Figure 3, the user has associated the degree 1 with *Whole milk* but only 0.3 with *Condensed milk*. The maximum of these two degrees is thus associated with their common sub-element *Condensed whole milk* in the closure.

The case of *Sweetened condensed milk* is different: the user has associated the degree 0.8 with *Milk* but has given a restriction on the more specific element *Condensed milk* (degree 0.3). As *Sweetened condensed milk* is a kind of *Condensed milk*, it inherits the degree associated with *Condensed milk*, that is 0.3.

In the case where an element elt of the hierarchy, that does not appear in the initial hierarchical fuzzy set, has several smallest super-elements that appear in the hierarchical fuzzy set with different degrees, associating the maximum of these degrees with elt in the closure is a choice that may be discussed. We distinguish two cases:

- if the hierarchical fuzzy set expresses preferences in a query, the choice of the maximum allows us not to exclude any possible answer (the possibility and the necessity degrees of matching can be higher). In real cases, the lack of answers to a query generally makes this choice preferable, because it consists in enlarging the query rather than restricting it. This is actually the case in our project;
- if the hierarchical fuzzy set represents an ill-known datum, the choice of the maximum allows us to preserve all the possible values of the datum, but it also makes the datum less specific. We chose this solution in order to homogenize the treatment of queries and data. In a way, it also participates in enlarging the query, as a less specific datum may share more common values with the query (the possibility degree of matching can thus be higher, although the necessity degree can decrease).

Computing the closure $clos(F)$ of a fuzzy set F defined on a domain $dom(F) \subset H$ has a complexity in $|H|.|dom(F)|^2$, provided that the comparison of two elements of the hierarchy can be done in constant time. Generally, the definition domain of F is limited to a few elements, so that the actual computing time remains moderate.

Complexity Analysis 1: The steps of the computing are the following:

There are $(|H| - |dom(F)|)$ elements in H that do not appear in $dom(F)$. The degree that is associated with them in $clos(F)$ thus has to be determined. For each element elt of these $(|H| - |dom(F)|)$ elements, one must:

- compare elt with each of the $|dom(F)|$ elements of F (there are $|dom(F)|$ comparisons), so as to determine the super-elements of elt in $dom(F)$. We denote S the set of super-elements of elt in $dom(F)$. We have: $|S| \leq |dom(F)|$. We consider that the comparison of two elements can be done in constant time.
- among the $|S|$ super-elements of elt in $dom(F)$, determine the most specific ones. Therefore, the $|S|$ super-elements must be compared to one other. In the worst case,

they will all be compared by two, which will require $C^2_{|S|} = \frac{|S|(|S|-1)}{2}$ comparisons. We denote $\tilde{S}$ the set of most specific super-elements of elt in $|dom(F)|$. We have: $|\tilde{S}| \leq |dom(F)|$;

– among the degrees associated, in F, with the $|\tilde{S}|$ most specific super-elements of elt, choose the greatest one. This maximum calculus is done by comparing the degree associated with one of the $|\tilde{S}|$ elements with the degrees of the other $(|\tilde{S}| - 1)$, and choosing the greater each time. There are thus $(|\tilde{S}| - 1)$ comparisons.

For each of the $(|H| - |dom(F)|)$ elements that do not appear in $dom(F)$, the number of comparisons that are computed is finally: $(|dom(F)| + \frac{|S|.(|S|-1)}{2} + |\tilde{S}| - 1)$.

The total number of comparisons that is computed is thus: $(|H| - |dom(F)|)(|dom(F)| + \frac{|S|.(|S|-1)}{2} + |\tilde{S}| - 1)$, which is majored by $(|H| - |dom(F)|)(|dom(F)| + \frac{|dom(F)|.(|dom(F)|-1)}{2} + |dom(F)| - 1) = \frac{1}{2}(|H| - |dom(F)|)(|dom(F)|^2 + 3|dom(F)| - 2)$.

We can note that:

- if $|dom(F)| = |H|$, that is, if F is already a closure defined on H, there is of course no operation to do;
- if $|dom(F)|$ is small compared to $|H|$, which is generally the case (in the project, $|dom(F)|$ is limited to 5), computing the closure is then linear in $|H|$;
- otherwise, computing the closure is polynomial. As $|dom(F)|$ and $(|H| - |dom(F)|)$ are majored by $|H|$, the complexity is in $O(|H|^3)$.

C. Comparisons of hierarchical fuzzy sets

The introduction of the concept of closure allows all the fuzzy sets that are defined on a given hierarchy to have the same definition domain (the whole hierarchy) and thus to be compared using the classical comparisons and operations between fuzzy sets.

Definition 6: Let F_1 and F_2 be two hierarchical fuzzy sets defined on the same hierarchy. Then:

1) $F_1 \subseteq F_2$ if $clos(F_1) \subseteq clos(F_2)$;
2) the possibility degree of matching between F_1 and F_2, $\Pi(F_1; F_2)$, is defined as $\Pi(clos(F_1); clos(F_2))$;
3) the necessity degree of matching between F_1 and F_2, $N(F_1; F_2)$, is defined as $N(clos(F_1); clos(F_2))$.

Example 2: The closures of the hierarchical fuzzy sets *1/Skim milk + 0.2/Milk* and *1/Milk + 0.5/Condensed milk* are represented in Figures 4 and 5. Their comparison shows that *1/Skim milk + 0.2/Milk* is included in *1/Milk + 0.5/Condensed milk* because the membership function of the former associates lower degrees with every element of the hierarchy.

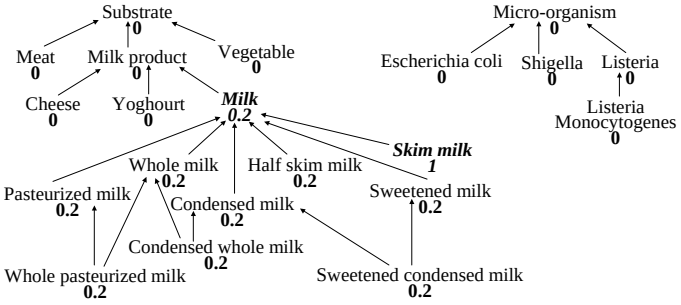


Fig. 4. Closure of the hierarchical fuzzy set *1/Skim milk + 0.2/Milk*

is included in:

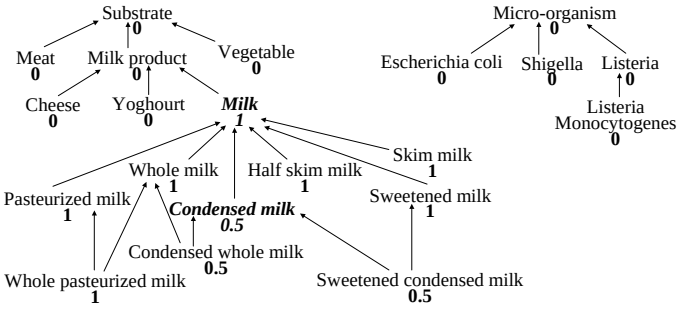


Fig. 5. Closure of the hierarchical fuzzy set *1/Milk + 0.5/Condensed milk*

D. Minimal fuzzy sets

In Section III-B, we saw that each hierarchical fuzzy set has an associated closure that is defined on the whole hierarchy. We now focus on the fact that two different hierarchical fuzzy sets, defined on the same hierarchy, can have the same closure, as in the following examples.

Example 3: The hierarchical fuzzy sets *Substrate$_1$ = 1/Milk* and *Substrate$_2$ = 1/Milk + 1/Skim milk* have the same closure: the degree 1 is associated with *Milk* and every more specific element, the degree 0 is associated with all the other elements of the hierarchy.

Example 4: The hierarchical fuzzy sets *Substrate*$_3$ = *1/Milk + 0.8/Whole milk + 1/Pasteurized milk* and *Substrate*$_4$ = *1/Milk + 0.8/Whole milk + 1/Whole pasteurized milk* have the same closure, represented in Figure 6.

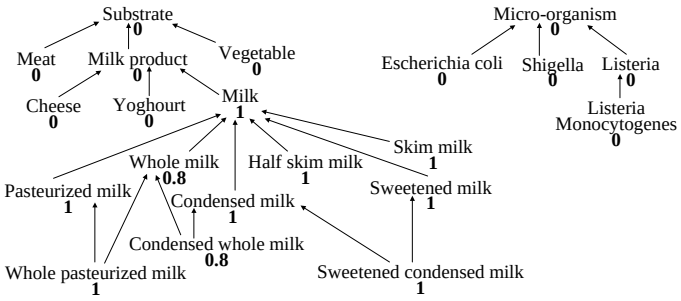


Fig. 6. Common closure of the hierarchical fuzzy sets *Substrate*$_3$ and *Substrate*$_4$

Such hierarchical fuzzy sets form equivalence classes with respect to their closures.

Definition 7: Two hierarchical fuzzy sets F_1 and F_2, defined on the same hierarchy, are said to be **equivalent** (denoted $F_1 \equiv F_2$) if and only if they have the same closure.

Property 1: Let F_1 and F_2 be two equivalent hierarchical fuzzy sets. If $elt \in dom(F_1) \cap dom(F_2)$ then $\mu_{F_1}(elt) = \mu_{F_2}(elt)$.

Proof 1: According to the definition of the closure of a hierarchical fuzzy set F (Definition 5), the closure of F preserves the degrees that are specified in F. As F_1 and F_2 have the same closure (by definition of the equivalence), an element that belongs to F_1 and F_2 necessarily has the same degree in both. □

We can note that *Substrate*$_2$ contains the same element as *Substrate*$_1$ with the same degree, and also one more element (*Skim milk*, with the degree 1). The degree associated with this additional element is the same as in the closure of *Substrate*$_2$. We say that the element *Skim milk* is *deducible* in *Substrate*$_2$.

Definition 8: Let F be a hierarchical fuzzy set, with $dom(F) = \{elt_1, ..., elt_j, ..., elt_n\}$, and F_{-j} the fuzzy set resulting from the restriction of F to the domain $dom(F) \setminus \{elt_j\}$. elt_j is **deducible** in F if $\mu_{clos(F_{-j})}(elt_j) = \mu_F(elt_j)$.

As a first intuition, we could say that removing a deducible element from a hierarchical fuzzy set allows one to eliminate redundant information. But an element being deducible in F does not necessarily mean that removing it from F will have no consequence on the closure: removing elt from F will not impact the degree associated with elt itself in the closure, but it may impact the degrees of the sub-elements of elt in the closure. For instance, the element *Pasteurized milk* is deducible in *Substrate*$_3$, according to Definition 8. Removing *1/Pasteurized milk* from *Substrate*$_3$ would not modify the degree of *Pasteurized milk* itself in the resulting closure, but it would modify the degree of its sub-element *Whole pasteurized milk* (which would have the degree 0.8 instead of 1). Thus, this remark leads us to the following definition of a minimal hierarchical fuzzy set.

Definition 9: In a given equivalence class (that is, for a given closure C), a hierarchical fuzzy set is said to be **minimal** if its closure is C and if none of the elements of its domain is deducible (here the term "minimal" does not have the meaning of cardinality).

The hierarchical fuzzy sets *Substrate*$_1$ and *Substrate*$_4$ are minimal (none of their elements is deducible), contrary to *Substrate*$_2$ and *Substrate*$_3$.

We have proposed an algorithm and its proofs, given below, to calculate a minimal fuzzy set. The proofs establish the following two properties.

Property 2: The stopping condition is always reached.

Property 3: The hierarchical fuzzy set obtained with this algorithm is minimal.

Algorithm 1:

Calculation of a minimal fuzzy set mnl having a given closure C

begin
  $mnl \leftarrow \emptyset$
  if ($clos(mnl) = C$)
  then
    stop (case where C is the hierarchical fuzzy set that associates the degree 0 with every element of the hierarchy)
  else
    let lin be an order such that each element of the hierarchy is examined after its super-elements (that is, a linear extension of the opposite order of that induced by the "kind of" relation)
    repeat
      $elt \leftarrow$ next element according to lin
      if ($\mu_{clos(mnl)}(elt) \neq \mu_C(elt)$)
      then
        $mnl \leftarrow mnl \cup \{elt\}$
        $\mu_{mnl}(elt) \leftarrow \mu_C(elt)$
      endif
    until $(clos(mnl) = C)$
  endif
end

Proof 2: Proof of Property 2

At the beginning of the algorithm, there are two possible cases:

- either the stopping condition is already satisfied;
- or the stopping condition is not satisfied: then the elements of the hierarchy start to be examined in the order lin (each element is examined after its super-elements). Let us process by induction to show that, after the n^{th} element is examined, every element elt among the first n elements that have already been examined satisfies: $\mu_{clos(mnl)}(elt) = \mu_C(elt)$. $n \in [1, N]$, where N (range of the last element that is examined before the algorithm stops) is at most equal to the number of elements of the hierarchy; N is smaller if the stopping condition is reached before all the elements are examined.

For $n = 1$: Before the first element is examined, mnl is empty and its closure associates the degree 0 with all the elements of the hierarchy. Let elt_1 be the first element that is examined. There are two possible cases:

1) elt has the degree 0 in C. We thus have $\mu_{clos(mnl)}(elt_1) = \mu_C(elt_1) = 0$. The algorithm directly goes to the next element;
2) the degree d associated with elt_1 in C is different from 0. In this case, elt_1 is added to mnl with the degree d. We thus have $\mu_{clos(mnl)}(elt_1) = \mu_C(elt_1) = d$.

After the first element is examined, this first element elt_1 always satisfies the condition: $\mu_{clos(mnl)}(elt_1) = \mu_C(elt_1)$.

Let us suppose that, after the n^{th} element is examined, each of the first n elements elt_1, ..., elt_i, ..., elt_n which have already been examined satisfies the condition: $\mu_{clos(mnl)}(elt_i) = \mu_C(elt_i)$. mnl associates a given degree x with the $(n+1)^{th}$ element elt_{n+1}. When elt_{n+1} is examined, there are two possible cases:

1) elt_{n+1} has the degree x in C. We thus have $\mu_{clos(mnl)}(elt_{n+1}) = \mu_C(elt_{n+1}) = x$. The algorithm directly goes to the next element. We still have $\forall i \in [1, n],\ \mu_{clos(mnl)}(elt_i) = \mu_C(elt_i)$ because mnl has not been changed;
2) the degree d_{n+1} associated with elt_{n+1} in C is different from x. In this case, elt_{n+1} is added to mnl with the degree d_{n+1}. We thus have $\mu_{clos(mnl)}(elt_{n+1}) = \mu_C(elt_{n+1}) = d_{n+1}$. This time, mnl has been changed by adding elt_{n+1}. Compared to each elt_i ($i \in [1, n]$), elt_{n+1} is either more specific, or not comparable, but elt_{n+1} cannot be a super-element of elt_i, because of the order lin. Therefore, adding elt_{n+1} in mnl does not change the degrees that are associated with $elt_1, \ldots, elt_i, \ldots, elt_n$ in the closure of mnl. Indeed, the degree associated with elt_i in the closure of mnl only depends on the super-elements (in the broad sense) of elt_i in mnl, according to the definition of the closure (Definition 5). We thus still have $\forall i \in [1, n],\ \mu_{clos(mnl)}(elt_i) = \mu_C(elt_i)$.

After the $(n+1)^{th}$ element is examined, each element elt among the first $n+1$ elements that have already been examined satisfies the condition: $\mu_{clos(mnl)}(elt) = \mu_C(elt)$.

We finally obtain, at most after the last element of the hierarchy has been examined: $\forall elt,\ \mu_{clos(mnl)}(elt) = \mu_C(elt)$, that is, the stopping condition $clos(mnl) = C$. □

Proof 3: Proof of Property 3

Let us process by induction to show that, for each iteration of the algorithm, mnl is minimal (with the meaning of Definition 9).

At the beginning, mnl is empty and thus minimal.

Let us suppose that $mnl = \{elt_1, \ldots, elt_i, \ldots, elt_k\}$ is minimal after the k^{th} iteration of the algorithm: each element elt_i in mnl is non-deducible (Definition 9). At the $(k+1)^{th}$ iteration, the algorithm adds to mnl the next element elt_{k+1} of the hierarchy (in the order lin) which does not have the same degree in the closure of mnl as in C ($\mu_{clos(mnl)}(elt_{k+1}) \neq \mu_C(elt_{k+1})$), that is, which is not deducible in mnl (Definition 8). mnl is modified by adding elt_{k+1}. We may thus wonder if the elements elt_i ($i \in [1,k]$) are still non-deducible in mnl. Because of the order lin, elt_{k+1} cannot be a super-element of elt_i ($i \in [1,k]$). Therefore, adding elt_{k+1} in mnl brings no change in the degrees associated with $elt_1, \ldots, elt_i, \ldots, elt_k$ in the closure of mnl. Indeed, the degree associated with elt_i in the closure of mnl only depends on the super-elements (in the broad sense) of elt_i in mnl, according to the definition of the closure (Definition 5). The elements elt_i ($i \in [1,k]$) are thus still non-deducible in mnl. After the $(k+1)^{th}$ iteration of the algorithm, mnl is minimal because all its elements are non-deducible. □

Property 4: The minimal fuzzy set is **unique** for a given closure.

Proof 4: Let F_1 and F_2 be two minimal fuzzy sets, with $F_1 \equiv F_2$ and $F_1 \neq F_2$. Note that we cannot have $dom(F_1) = dom(F_2)$, otherwise F_1 and F_2 would not be different (Property 1). Let elt be one of the *most general* elements (with the meaning of the "kind of" relation) that belong to $(dom(F_1) \cup dom(F_2)) \setminus (dom(F_1) \cap dom(F_2))$. Figure 7 shows the possible localization of elt and its super-elements.

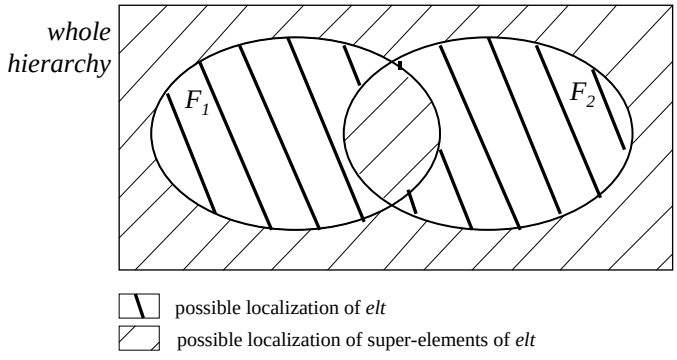


Fig. 7. Possible localization of elt and its super-elements

elt thus belongs to $dom(F_1)$ or to $dom(F_2)$, but not to both, and it has no super-element that satisfies this condition: every super-element of elt necessarily belongs either to $dom(F_1)$ and $dom(F_2)$, or neither to $dom(F_1)$ nor to $dom(F_2)$.

Let F_x be the hierarchical fuzzy set (F_1 or F_2) whose domain contains elt. The other one is denoted F_y. There are two possible cases:

- $dom(F_1) \cap dom(F_2)$ contains no super-element of elt. As $dom(F_y)$ contains neither elt nor any of its super-elements, we have $\mu_{clos(F_y)}(elt) = 0$. On the contrary, as $dom(F_x)$ contains elt, we have $\mu_{clos(F_x)}(elt) = \mu_{F_x}(elt)$ which is necessarily different from 0, otherwise F_x would not be minimal because elt would be deducible in F_x. As F_1 and F_2 do not have the same closure, they are not equivalent, which contradicts our hypothesis;
- $dom(F_1) \cap dom(F_2)$ contains one or more super-elements of elt. Let S_{elt} be the set of these super-elements and $E_{elt} = \{elt_1, ..., elt_j, ..., elt_n\}$ the set of most specific one(s) among them (with the meaning of the "kind of" relation). For each $elt_j \in E_{elt}$, we have $\mu_{F_1}(elt_j) = \mu_{F_2}(elt_j)$ according to Property 1. As $dom(F_y)$ does not contain elt but contains S_{elt}, we have $\mu_{clos(F_y)}(elt) = max_{1 \leq j \leq n}(\mu_{F_y}(elt_j))$ according to Definition 5. On the contrary, as $dom(F_x)$ contains elt, we have $\mu_{clos(F_x)}(elt) = \mu_{F_x}(elt)$ which is necessarily different from $max_{1 \leq j \leq n}(\mu_{F_y}(elt_j))$, otherwise F_x would not be minimal because elt would be deducible in F_x. As F_1 and F_2 do not have the same extension, they are not equivalent, which contradicts our hypothesis. □

Example 5: Let C be the closure represented in Figure 6. The minimal fuzzy set mnl is obtained as follows:

Initially, mnl is empty. Its closure is the hierarchical fuzzy set that associates the degree 0 with each element of the hierarchy. We test if this closure is C. The answer is no, as not all the elements have the degree 0 in C. We thus traverse the hierarchy using an order such that each element is examined after its super-elements.

We first examine, for instance, *Substrate*. It has the same degree 0 in the closure of mnl and in C. We continue with *Meat*, *Milk product*, *Vegetable*, *Cheese* and *Yoghourt*, which also have the same

degree in the closure of mnl as in C.
Then we examine *Milk*. It has the degree 0 in the closure of mnl, whereas its degree is 1 in C. *Milk* is thus added to mnl, with the degree 1. The closure of mnl is now the hierarchical fuzzy set that associates the degree 1 with *Milk* and with the sub-elements of *Milk*, and 0 with all the other elements of the hierarchy, which is different from C. We thus go on traversing the hierarchy.
Pasteurized milk has the same degree 1 in the closure of mnl and in C. We thus continue.
Whole milk has the degree 1 in the closure of mnl but the degree 0.8 in C. *Whole milk* is thus added to mnl, with the degree 0.8. The closure of mnl is now the hierarchical fuzzy set that associates the degree 0.8 with *Whole milk* and with the sub-elements of *Whole milk* (*Whole pasteurized milk* and *Condensed whole milk*), the degree 1 with the other milks (*Milk*, *Pasteurized milk*, *Condensed milk*, etc.) and 0 with the other elements of the hierarchy, which is still different from C. We go on traversing the hierarchy. We examine *Condensed milk*, then *Half skim milk*, *Sweetened milk* and *Skim milk*, which all have the same degree 1 in the closure of mnl as in C.
Whole pasteurized milk has the degree 0.8 in the closure of mnl but the degree 1 in C. It is added to mnl with the degree 1. The closure of mnl is now the hierarchical fuzzy set that associates the degree 0.8 with *Whole milk* and its sub-element *Condensed whole milk*, the degree 1 with all the other milks and 0 with the rest of the hierarchy, which is equal to C. The algorithm stops.

We finally obtain mnl = *1/Milk + 0.8/Whole milk + 1/Whole pasteurized milk*, which corresponds to *Substrate4*.

Computing the minimal fuzzy set mnl of a given closure C defined on a hierarchy H has a complexity in $|H|.|dom(mnl)|^2$.

Complexity Analysis 2: Computing the minimal fuzzy set requires to examine each element elt of H, using an order lin that conforms to Algorithm 1, to determine if $\mu_{clos(mnl_i)}(elt) = \mu_C(elt)$ (where mnl_i is the current state of calculus of mnl) and add elt to mnl_i if this equality is not satisfied:

- determining if $\mu_{clos(mnl_i)}(elt) = \mu_C(elt)$ requires to calculate the closure of mnl_i for the element elt only (see Complexity Analysis 1). As the number of elements in mnl_i is always majored by $|dom(mnl)|$, the complexity of this operation is always inferior to: $\frac{1}{2}(|dom(mnl)|^2 + 3|dom(mnl)| - 2)$;
- adding elt to mnl (if $\mu_{clos(mnl_i)}(elt) \neq \mu_C(elt)$) is done in constant time.

For the whole hierarchy H, the complexity is thus inferior to: $\frac{1}{2}|H|(|dom(mnl)|^2 + 3|dom(mnl)| - 2)$.

If $|dom(mnl)|$ is small compared to $|H|$, computing the minimal fuzzy set is thus linear in $|H|$. In the extreme case where $|dom(mnl)| = |H|$, we obtain: $\frac{1}{2}(|H|^3 + 3|H|^2 - 2|H|)$. Computing the minimal fuzzy set is then polynomial in $|H|$.

IV. Generalization of a hierarchical fuzzy set

In this section, we propose a complementary solution to the lack of answers to a query, used when the user wants to retrieve complementary answers close to his initial query. The hierarchical fuzzy set that represents the user's preferences is replaced by a more general one, with the meaning of the inclusion relation extended to hierarchical fuzzy sets.

Different approaches have been proposed in the literature in order to introduce tolerance in the querying. In [18], a fuzzy operator based on proximity relation is proposed to weaken fuzzy predicates in a query, but it concerns numerical domains and cannot be applied to predicates defined on a hierarchically organized domain. Tolerant fuzzy pattern matching [15] uses a similarity relation between terms to enlarge the preferences, but it does not take into account the case of hierarchically organized domains. For instance, terms may be added to the support of the fuzzy set in the enlargement mechanism, but more specific terms than these ones may stay outside of it, which is a major drawback for hierarchical domains. Other measures have been introduced to evaluate how close to each other two fuzzy graphical representations are [19] or taking into account preexistent similarity relations [20], [21]. In studies concerning information retrieval non-limited to exact answers (see [22]–[24]), searching for approximate answers has been managed in two ways: modifying the datum so that it may satisfy the query, or modifying the query so that it may be satisfied by the datum. Our work conforms to the latter approach, however we are in the context of a database application, and not a corpus of textual documents which is a different concern.

More than a unique solution, we propose a methodology in order to generalize a hierarchical fuzzy set expressing preferences.

A. Elementary generalization of a hierarchical fuzzy set

The elementary generalization of a hierarchical fuzzy set consists in creating, given a hierarchical fuzzy set F, a more general hierarchical fuzzy set F_g, with the meaning of the inclusion relation defined in Section III-C. To obtain F_g, an element elt_g is added to F, elt_g being a super-element of an element $elt \in dom(F)$. We have defined this operation to be as flexible as possible.

Definition 10: An **elementary generalization** of a hierarchical fuzzy set F is an operation that creates from F a hierarchical fuzzy set F_g obtained as follows.
Let elt be an element of $dom(F)$ and elt_g a super-element of elt, satisfying the condition: $\nexists elt' \in dom(F)\ (elt_g \leq elt')$. That is to say, elt_g may neither be an element of $dom(F)$ nor be more specific than any element of $dom(F)$.
F_g is obtained by adding elt_g to F with a given degree denoted d_g. F_g is thus defined by:
$\begin{cases} dom(F_g) = dom(F) \cup \{elt_g\} \\ \mu_{F_g}(elt_g) = d_g. \end{cases}$

Property 5: F_g is more general than F, with the meaning of the inclusion relation extended to hierarchical fuzzy sets.

Proof 5: We must show that, for each element $elem$ of the hierarchy, we have: $(\mu_{clos(F_g)}(elem) \geq \mu_{clos(F)}(elem))$.

Let $E_{elem} = \{elem_1, \ldots, elem_n\}$ be the set of smallest super-elements (in the broad sense) of $elem$ in $dom(F)$. According to the definition of the closure (Definition 5), $\mu_{clos(F)}(elem)$ only depends on E_{elem}. Let E_{elem_g} be the set of smallest super-elements of $elem$ in $dom(F_g)$. $\mu_{clos(F_g)}(elem)$ only depends on E_{elem_g}. We will show that E_{elem_g} is equal, either to E_{elem}, or to $E_{elem} \cup \{elt_g\}$, and that the inequality $\mu_{clos(F_g)}(elem) \geq \mu_{clos(F)}(elem)$ is satisfied in both cases.

As $dom(F_g) = dom(F) \cup \{elt_g\}$ and that elt_g cannot be a sub-element of an element of $dom(F)$ (Definition 10), *a fortiori* elt_g cannot be a sub-element of an element of $E_{elem} \subseteq dom(F)$. There are thus two possible cases:

- elt_g is a super-element of one ore more elements of E_{elem}. Thus it cannot be itself a smallest super-element of $elem$ in $dom(F_g)$: $elt_g \notin E_{elem\_g}$. Therefore we have $E_{elem\_g} = E_{elem}$ and $\mu_{clos(F_g)}(elem) = \mu_{clos(F)}(elem)$;
- elt_g is not comparable with any element of E_{elem} (or E_{elem} is empty). In this case:
  - either elt_g is not a super-element of $elem$. Then $E_{elem\_g} = E_{elem}$ and $\mu_{clos(F_g)}(elem) = \mu_{clos(F)}(elem)$;
  - or elt_g is a super-element of $elem$. Then $E_{elem\_g} = E_{elem} \cup \{elt_g\}$ (elt_g is necessarily a smallest super-element of $elem$ in $dom(F_g)$ because it is not comparable with the elements of E_{elem}). There are two possible cases:
    - if E_{elem} is empty, $\mu_{clos(F)}(elem) = 0$ and $\mu_{clos(F_g)}(elem) = \mu_{F_g}(elt_g) \geq 0$;
    - if E_{elem} is not empty, $\mu_{clos(F)}(elem) = max(\mu_F(elem_1), \ldots, \mu_T(elem_n))$ and $\mu_{clos(F_g)}(elem) = max(\mu_{F_g}(elem_1), \ldots, \mu_{F_g}(elem_n), \mu_{F_g}(elt_g)) = max(\mu_F(elem_1), \ldots, \mu_F(elem_n), \mu_{F_g}(elt_g)) \geq \mu_{clos(F)}(elem)$.

We thus have for each $elem$:
$\mu_{clos(F_g)}(elem) \geq \mu_{clos(F)}(elem)$. □

Example 6: Let F be the following hierarchical fuzzy set: *F = 1/Condensed whole milk + 0.5/Cheese*.
For *elt = Condensed whole milk*, elt_g = *Milk* and d_g = 0.2, we obtain:
F_g = 1/Condensed whole milk + 0.5/Cheese + 0.2/Milk.

B. Generalization rule

The elementary generalization defined above will be used as a basis for the definition of a (non-elementary) generalization, obtained by applying to F several elementary generalizations: for each element of F, a set of more general elements may be added to F.

Therefore, several questions have to be decided: (i) in which order will the elements of F be considered, as this order may affect the result ? (ii) which more general elements may be added to F ?

Comment citer ce document :
Thomopoulos, R., Buche, P., Haemmerle, O. (2006). Fuzzy sets defined on a hierarchical domain.
IEEE Transactions on Knowledge and Data Engineering, 18 (10), 1397-1410. DOI : 10.1109/TKDE.2006.161

(iii) how will the degree associated with each added element be determined ?

These questions arise from issues frequently found in literature about similarity, in different contexts concerning non-fuzzy or non-hierarchical values, or using additional knowledge as in linguistic issues. Questions (ii) and (iii) are linked to the notion of distance between concepts [25]–[28]. Question (iii) also impacts the classification of the results to be obtained [15]. Question (i) concerns possible conflicts between elements of F having common super-elements added to F, with an antagonism about the choice of the degrees to be associated with these super-elements.

The notion of generalization rule formalizes these elements.

Definition 11: A **generalization rule** R_g is a 3-tuple $(ord,\ gen,\ calc)$, where:

- ord is a total traversal order through the elements of a hierarchical fuzzy set F, defined on a hierarchy H;
- gen is an application that associates, with each element elt in $dom(F)$, a set of more general elements in H;
- $calc$ is an application that associates a degree between 0 and 1 with each pair (elt, elt_g) such that $elt \in dom(F)$ and $elt_g \in gen(elt)$.

Example 7:

- ord may be, for instance, an order through the elements of F by decreasing degrees. This choice allows one to generalize in priority the elements of F that have the higher degrees, that is, the elements for which the user has expressed the higher preference;
- $gen(elt)$ may be, for instance, the set of smallest super-elements of elt in the hierarchy;
- $calc$ may, for instance, associate with elt_g half of the degree of the generalized element elt. This choice allows one to retrieve in priority the values specified by the user.

Each element of F does not necessarily have a more general element that may be added to F for the generalization operation: as we saw previously in Section IV-A, this more general element must satisfy a condition. Here we define the notion of generalizable element of F, according to a given generalization rule.

Definition 12: Let F be a hierarchical fuzzy set. An element elt of $dom(F)$ is said to be **generalizable** in F, according to a generalization rule R_g, if elt has a more general element elt_g in $gen(elt)$ that satisfies the condition: $\nexists elt' \in dom(F)\ (elt_g \leq elt')$.

Example 8: Let F be the following hierarchical fuzzy set: F = *1/Whole milk + 0.5/Milk*, and $st(elt)$ the set of smallest super-elements of elt.
The element *Whole milk* is not generalizable in F because st*(Whole milk)* $= \{$*Milk*$\}$, and *Milk* is already in F.
The element *Milk* is generalizable because st*(Milk)* $= \{$*Milk product*$\}$, and *Milk product* is not in F nor has a super-element in F.

C. Elementary generalization according to a generalization rule

The elementary generalization presented here is an operation that conforms to Definition 10, restricted by a generalization rule R_g.

Definition 13: An **elementary generalization according to a generalization rule** R_g is an elementary generalization of a hierarchical fuzzy set F, such that:

- the element $elt \in dom(F)$ is chosen as being the first generalizable element of F, according to the order ord. It is denoted elt_0;
- elt_g is a smallest super-element of elt_0 in $gen(elt_0)$;
- d_g is defined by: $d_g = calc(elt_0, elt_g)$.

Example 9: Let R_g be the generalization rule proposed in Example 7 and F the following hierarchical fuzzy set:
F = *1/Whole milk + 0.8/Half skim milk + 0.2/Yoghourt*.
All the elements of F are generalizable, and the first one according to the order ord (i.e. by decreasing degrees) is *Whole milk*. The elementary generalization of F according to R_g is thus the following hierarchical fuzzy set:
F_g = *1/Whole milk + 0.8/Half skim milk + 0.2/Yoghourt + 0.5/Milk*.

D. Generalization of a hierarchical fuzzy set, according to a generalization rule

The (non-elementary) generalization of F that we define here consists in applying successively several elementary generalizations, according to a given generalization rule, to the **minimal** fuzzy set that is equivalent to F. We chose to generalize the minimal fuzzy set, and not F itself, because we consider that different equivalent fuzzy sets expressing the user's

query and bringing the same answers, should have the same generalization that will bring the same additional answers. This is guaranteed by the use of the minimal fuzzy set.

Definition 14: The **generalization of a hierarchical fuzzy set** F, according to a generalization rule R_g, is an operation that provides a hierarchical fuzzy set F_g obtained as follows:

- we call 0-degree generalization of F, denoted F_0, the minimal fuzzy set that is equivalent to F;
- let F_n be the n-degree generalization of F:
  - if there exists an element of $dom(F_0) \subseteq dom(F_n)$ generalizable in F_n according to R_g, then F_{n+1} is obtained by an elementary generalization of F_n according to R_g, in which elt_0 is the first element of $dom(F_0) \subseteq dom(F_n)$, with the meaning of the order ord, generalizable in F_n;
  - if not, the generalization of F is the fuzzy set $F_g = F_n$.

Property 6: The degree n such that $F_g = F_n$ is finite.

Proof 6: Let GEN be the set of elements that belong to the image, through gen, of the set of elements of $dom(F_0)$: $GEN = \bigcup_{elt \in dom(F_0)} gen(elt)$.

The element elt_g^n added to F_n to obtain F_{n+1} belongs to the set:
$E_n = \{elt' \in GEN \mid \nexists elt \in dom(F_n)\ (elt' \leq elt)\}$. According to Definition 14, $F_g = F_n$ is obtained when E_n is empty.

The element elt_g^{n+1} added to F_{n+1} to obtain F_{n+2} belongs to the set: $E_{n+1} =$
$\{elt' \in GEN \mid \nexists elt \in dom(F_{n+1})\ (elt' \leq elt)\} =$
$\{elt' \in GEN \mid \nexists elt \in (dom(F_n) \cup \{elt_g^n\})\ (elt' \leq elt)\}$
$\subseteq E_n$.

E_{n+1} contains at least one less element than E_n: elt_g^n, which does not satisfy the condition $(\nexists elt \in (dom(F_n) \cup \{elt_g^n\})\ (elt_g^n \leq elt)\}$. We thus have: $card(E_{n+1}) < card(E_n)$. As $card(E_n)$ is strictly decreasing with n, E_n is empty for n at most equal to $card(E_0)$. The degree n such that $F_g = F_n$ is thus finite. □

Property 7: The fuzzy set F_g, obtained by the generalization of F, is more general than F, with the meaning of the inclusion relation extended to hierarchical fuzzy sets.

Proof 7: Let us process by induction to show that, for each n, we have:
for each element elt of the hierarchy, $\mu_{clos(F_n)}(elt) \geq \mu_{clos(F)}(elt)$.

For $n = 0$, we have: $\forall elt,\ \mu_{clos(T_0)}(elt) = \mu_{clos(F)}(elt)$, because $F \equiv F_0$.

Let us suppose that: $\forall elt,\ \mu_{clos(F_n)}(elt) \geq \mu_{clos(F)}(elt)$. As F_{n+1} is obtained by an elementary generalization of F_n, we have: $\forall elt,\ \mu_{clos(F_{n+1})}(elt) \geq \mu_{clos(F_n)}(elt)$ (Proposition 5). Therefore: $\forall elt,\ \mu_{clos(F_{n+1})}(elt) \geq \mu_{ext(F)}(elt)$. □

Example 10: Let R_g be the generalization rule proposed in Example 7 and F the following hierarchical fuzzy set:
F = 1/Whole milk + 1/Condensed whole milk + 0.8/Half skim milk + 0.2/Yoghourt.

- F_0, the minimal fuzzy set that is equivalent to F, is the following:
  F_0 = 1/Whole milk + 0.8/Half skim milk + 0.2/Yoghourt;
- the first generalizable element of F_0, in the order ord, is *Whole milk*, whose generalization provides F_1:
  F_1 = 1/Whole milk + 0.8/Half skim milk + 0.2/Yoghourt + 0.5/Milk;
- the first element of $dom(F_0)$ generalizable in F_1 is *Yoghourt*, whose generalization provides F_2:
  F_2 = 1/Whole milk + 0.8/Half skim milk + 0.2/Yoghourt + 0.5/Milk + 0.1/Milk product;
- there is no element of $dom(F_0)$ generalizable in F_2, so $F_g = F_2$.

V. Experimental evaluation of the proposed methods

Since 1999, our team has been involved in the Sym'Previus national project, which brings together industrial and academic partners to build a tool for the analysis of microbiological risks in food products (http://www.symprevius.org).

We firstly describe the system architecture in Section V-A. Section V-B proposes an experimental evaluation of the closure and generalization methods presented in this paper.

A. The system architecture

The risk analysis tool includes a database querying system called MIEL++[1], available on the Internet, that queries three databases: a relational

[1]acronym for the French translation of Enlarged Querying Engine

Comment citer ce document :
Thomopoulos, R., Buche, P., Haemmerle, O. (2006). Fuzzy sets defined on a hierarchical domain.
IEEE Transactions on Knowledge and Data Engineering, 18 (10), 1397-1410. DOI : 10.1109/TKDE.2006.161

database which contains the stable part of the information [13], a conceptual graph knowledge base which contains the weakly-structured part of the information [29], [30], and an XML base filled with data semi-automatically extracted from the Web [14], [31]. Figure 8 illustrates the system architecture.

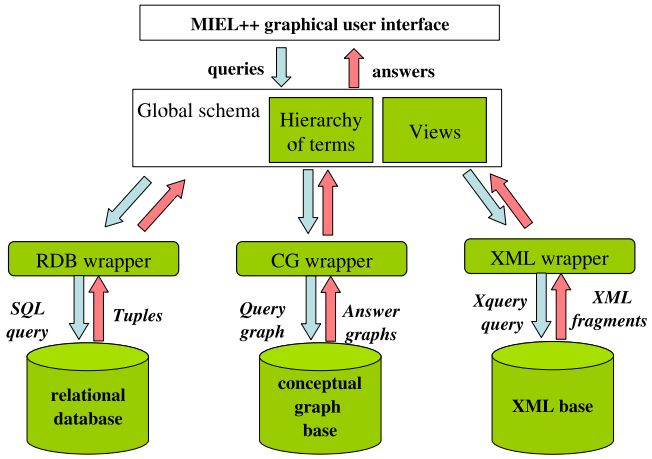


Fig. 8. The system global architecture

The vocabulary used to represent the data in the three databases, as well as to express queries in the retrieval system, is organized into a hierarchy of terms that corresponds to the taxonomies used by our biologist partners to represent classifications of microorganisms, food substrates and technological processes. The formalisms used in the three databases, as well as the MIEL++ query language, have been extended to be able to represent respectively ill-known data and flexible queries as fuzzy sets defined on a hierarchical domain.

In the MIEL++ system, a query is composed of a set of projection attributes and a set of selection criteria of the form $< attribute,\ value >$, where $value$ can be a hierarchical fuzzy set (see [13] for more details). It is expressed in a given view, which corresponds to a virtual table that brings together with sense all the attributes needed by the user.

The query expressed by the user through the GUI is sent to the three databases, and therefore translated into three different formalisms. In the case of the relational subsystem for instance, which gathers more than 10.000 data entries that correspond to scientific data from about 700 publications in predictive microbiology, it is translated into a SQL query. The $select$ clause is determined by the view in which the user expresses the query. The $where$ clause is determined both by the view and by the user's selection criteria. If the user's selection criteria contain hierarchical fuzzy sets, the closures of these fuzzy sets are computed and taken into account in the SQL query.

The MIEL++ system is written in Java language. A MIEL++ query is executed using a three-tier process architecture.

B. Evaluation of the closure and generalization methods

The evaluation was made in collaboration with microbiologist experts. The methodology we used to evaluate the proposed methods followed five steps:

1) definition of the quantity and thematic repartition of the data accessed by the test queries, so that the results are significant;
2) definition of the form of the test queries, so that the results are interpretable;
3) definition of a set of test queries that satisfy the previous points;
4) execution of the set of test queries;
5) analysis of the results.

In the following we describe the procedure step by step.

1) The significance conditions put on the test queries were, firstly, that they cover at least ten percent of the database entries, and secondly, that they cover all branches of the hierarchy of terms.
2) The interpretability conditions were of two kinds:
   - each test query should be executed in three forms: (i) as a standard query; (ii) with the computing of the closures of the selection criteria values; (iii) with the computing of the closures of the generalized selection criteria values;
   - what we mean by "standard" query is a query in which the selection criteria values are not fuzzy, so that there is no possible confusion in the interpretation of the degrees associated with the results: these degrees are due to the generalization method used in form (iii) of the query, and not to the user's preferences in form (i) of the query. Moreover in "standard" queries the closures are not computed, *i.e.* the sub-elements of the elements mentioned

in the selection criteria values are not taken into account.

3) Seven test queries were defined, with the following $< attribute,\ value >$ criteria:
   - <Food product, 1/Shell-fish>;
   - <Food product, 1/Cheese>;
   - <Food product, 1/Cheese> and <Microorganism, 1/Listeria>;
   - <Food product, 1/Egg>;
   - <Food product, 1/Potted meat> and <Microorganism, 1/Listeria>;
   - <Food product, 1/Salad> and <Microorganism, 1/Listeria>;
   - <Food product, 1/Fresh meat>.

   The parameters used in the generalization method are:
   - ord is by decreasing degrees;
   - $gen(elt)$ is the set of smallest super-elements of elt in the hierarchy;
   - $calc$ associates with elt_g the degree of elt minus 0.2 (or 0 if the result is negative).
4) The execution of the set of test queries gave the results presented in table I.
5) The analysis of the results led to the following conclusions.
   The closure results were considered as pertinent exact answers by the experts. They provide 99 percent of the total number of exact answers. The evaluation results are thus excellent for the closure method.
   Among the generalization results, the answers that are judged pertinent by the experts (80 percent) have the higher matching degrees, that go from 0.8 to 0.6, whereas the answers that are judged non-pertinent (20 percent) have degrees that go from 0.6 to 0.2. An essential constatation is that the value 0.6 can thus be considered as a threshold above which results are classified as pertinent, and below which results are classified as non-pertinent by the experts. The evaluation results are thus also very good for the generalization method, as:
   - pertinent results can be clearly identified using their matching degrees;
   - generalization results bring a big amount of complementary results (56 percent of the total number of pertinent results).

TABLE I
EVALUATION OF THE CLOSURE AND GENERALIZATION METHODS

Selection criteria	Number of exact answers with standard querying	Number of exact answers obtained with closure	Number of answers obtained with generalization judged pertinent (and noted degree)	Number of answers obtained with generalization judged non-pertinent (and noted degree)
Food product = 1/Shell-fish	0	4	66 (degree 0.8)	0
Food product = 1/Cheese	5	152	267 (degree 0.8)	0
Food product = 1/Cheese, Microorganism = 1/Listeria	0	53	87 (degree 0.8)	0
Food product = 1/Egg	0	16	10 (degree 0.8)	87 (degree 0.4 to 0.2)
Food product = 1/Potted meat, Microorganism = 1/Listeria	0	33	44 (degree 0.8)	63 (degree 0.6 to 0.4)
Food product = 1/Salad, Microorganism = 1/Listeria	0	17	25 (degree 0.8 to 0.6)	7 (degree 0.6 to 0.2)
Food product = 1/Fresh meat	0	217	136 (degree 0.8)	0

VI. CONCLUSION

Whereas in classic fuzzy sets, all the elements are on the same level and are associated with a degree explicitly defined, this is not necessarily the case in hierarchical fuzzy sets because several levels of detail exist in the hierarchy, and the hierarchical links between the elements have to be taken into account.

In our work, the hierarchical links are defined by the "kind of" relation. The membership of an element in a fuzzy set has consequences on the membership of its sub-elements in this fuzzy set. We thus define, as a first main contribution of this paper, the notion of hierarchical fuzzy set, that may be defined on a part of a hierarchy (for a given level of detail) and the notion of closure of a hierarchical fuzzy set, that is explicitly defined on the whole hierarchy, using the links between the elements that compose the hierarchy. Hierarchical fuzzy sets that have the same closure define equivalence classes, and each class has a unique particular representative, called minimal fuzzy set.

Minimal fuzzy sets are used as a basis to define the generalization of a hierarchical fuzzy set, which is the second main contribution of this paper. The methodology that we propose aims at enlarging the preferences expressed by a user in a query and represented as a hierarchical fuzzy set, in order to obtain pertinent complementary answers.

These results have been applied within the information system of the Sym'Previus project, dedicated to predictive microbiology. The Sym'Previus information system has been in production since the beginning of 2004 and may be consulted by users from research or industry by means of a subscription. As shown in the last section of this paper, the plus-value provided by the closure and generalization methods has been quantified and represents an important part of the pertinent answers.

We expect these results to be useful in new contexts: firstly, for flexible query answering in formalisms that do not originally handle a domain ontology; secondly, in other research fields that could benefit from flexible generalization / specialization methods, like knowledge discovery that hardly uses precision levels described by domain ontologies in learning processes.

We are now working on the optimization of the algorithms that are used to compare hierarchical fuzzy sets, which are currently based on the closures of the hierarchical fuzzy sets. We are considering a solution based on the use of minimal fuzzy sets. Another aspect of our current research, in the continuation of this paper, concerns the introduction of viewpoints in the considered hierarchies. An important point will also be to extend our results, in a meaningful way, to other sorts of relations.

REFERENCES

[1] P. Bosc, L. Liétard, and O. Pivert, "Soft querying, a new feature for database management system," in *Proceedings DEXA'94 (Database and EXpert system Application), Lecture Notes in Computer Science*, vol. 856. Springer-Verlag, 1994, pp. 631–640.

[2] P. Bosc and H. Prade, "An introduction to the fuzzy set and possibility theory-based treatment of soft queries and uncertain or imprecise databases," in *Uncertainty and Management in Information Systems: From Needs to Solutions*, A. Motro and P. Smets, Eds. Kluwer Academic Publishers, 1997, pp. 285–324.

[3] L. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, pp. 338–353, 1965.

[4] ——, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets and Systems*, vol. 1, pp. 3–28, 1978.

[5] J. Fargues, "CG information retrieval using linear resolution, generalization and graph splitting," in *Proceedings of the Fourth Annual Workshop on Conceptual Graphs*, J. Nagle and T. Nagle, Eds., Detroit, USA, August 1989.

[6] A. Bidault, C. Froidevaux, and B. Safar, "Repairing queries in a mediator approach," in *14th European Conference on Artificial Intelligence*, Berlin, 2000, pp. 406–410.

[7] Y. Loiseau, M. Boughanem, and H. Prade, "Evaluation of term-based queries using possibilistic ontologies," in *Soft Computing for Information Retrieval on the Web*, E. Herrera-Viedma, G. Pasi, and F. Crestani, Eds. Springer-Verlag, 2005.

[8] J. Rossazza, D. Dubois, and H. Prade, "A hierarchical model of fuzzy classes," in *Fuzzy and Uncertain Object-Oriented Databases: Concepts and Models*, ser. Advances in Fuzzy Systems - Applications and Theory, R. De Caluwe, Ed., vol. 13. World Scientific, 1998, pp. 21–61.

[9] S. Miyamoto and K. Nakayama, "Fuzzy information retrieval based on a fuzzy pseudothesaurus," *IEEE Transactions of Systems, Man and Cybernetics*, vol. 16, no. 2, pp. 278–282, 1986.

[10] M. De Cock, S. Guadarrama, and M. Nikravesh, "Fuzzy thesauri for and from the www," in *Soft Computing for Information Processing and Analysis*, M. Nikravesh, L. Zadeh, and J. Kacprzyk, Eds. Springer-Verlag, 2004, pp. 275–284.

[11] R. Thomopoulos, P. Buche, and O. Haemmerlé, "Different kinds of comparisons between fuzzy conceptual graphs," in *Proceedings of the 11th International Conference on Conceptual Structures, ICCS'2003, Lecture Notes in Artificial Intelligence*. Dresden, Germany: Springer, July 2003, pp. 54–68.

[12] R. Thomopoulos, "Représentation et interrogation élargie de données imprécises et faiblement structurées," Ph.D. dissertation, Institut national agronomique Paris-Grignon, France, 2003.

[13] P. Buche, C. Dervin, O. Haemmerlé, and R. Thomopoulos, "Fuzzy querying of incomplete, imprecise and heterogeneously structured data in the relational model using ontologies and rules," *IEEE Transactions on Fuzzy Systems*, vol. 13, no. 3, pp. 373–383, June 2005.

[14] P. Buche, J. Dibie-Barthélemy, O. Haemmerlé, and M. Houhou, "Towards flexible querying of xml imprecise data in a dataware house opened on the web," in *Proceedings of the 6th International Conference Flexible Querying Answering Systems (FQAS'2004)*. Lyon, France: Lecture Notes in AI #3055, Springer, June 2004, pp. 28–40.

[15] D. Dubois and H. Prade, "Tolerant fuzzy pattern matching: an introduction," in *Fuzziness in Database Management Systems*, P. Bosc and J. Kacprzyk, Eds. Heidelberg: Physica-Verlag, 1995, pp. 42–58.

[16] ——, *Possibility Theory - An Approach to Computerized Processing of Uncertainty*. New York: Plenum Press, 1988.

[17] H. Prade, "Lipski's approach to incomplete information data bases restated and generalized in the setting of Zadeh's possibility theory," *Information Systems*, vol. 9, no. 1, pp. 27–42, 1984.

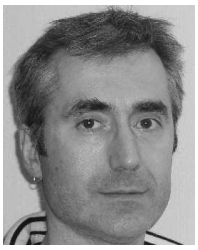
[18] P. Bosc, A. HadjAli, and O. Pivert, "Fuzzy closeness relation as a basis for weakening fuzzy relational queries," in *Proceedings of the 6th International Conference on Flexible Query-Answering Systems (FQAS'04)*, Lyon, France, June 2004, pp. 41–53.

[19] P. Bosc and O. Pivert, "On representation-based querying of databases containing ill-known values," in *Proceedings of the 10th International Symposium on Methodologies for Intelligent Systems (ISMIS'1997)*, 1997, pp. 477–486.

[20] R. George, A. Yazici, B. P. Buckles, and F. Petry, *Modeling Impreciseness and Uncertainty in the Object-Oriented Data Model - A Similarity-Based Approach*. Advances in Fuzzy

systems- Applications and Theory, Vol. 13, World scientific, 1997, pp. 63–95.
[21] R. George, R. Srikanth, B. P. Buckles, and F. Petry, *An approach to modelling impreciseness and uncertainty in the object-oriented data model*. John Wiley and Sons, Inc., 1997, pp. 325–337.
[22] C. Van Rijsbergen, "A non-classical logic for information retrieval," *The Computer Journal*, vol. 29, no. 6, pp. 481–485, 1986.
[23] J. Nie, "An outline of a general model for information retrieval," in *Proceedings of the 11th Annual ACM Conference on Research and Development in Information Retrieval*, Grenoble, France, 1988.
[24] D. Genest and M. Chein, "A content-search information retrieval process based on conceptual graphs," *to appear in Knowledge and Information Systems*, 2004.
[25] Z. Wu and M. Palmer, "Verb semantics and lexical selection," in *Proceedings of the 32nd Annual Meating of the Association for Computanional Linguistics*, 1994.
[26] D. Lin, "An information-theoretic definition of similarity," in *Proceedings of the 15th International Conference on Machine Learning, ICML'98*, Madison, Wisconsin, USA, 1998, pp. 296–304.
[27] P. Resnik, "Semantic similarity in a taxonomy: An information-based measure and its application to problem of ambiguity in natural language," *Journal of Artificial Intelligence Research*, no. 11, pp. 95–130, 1999.
[28] T. Andreasen, J. Nilsson, and H. Thomsen, "Ontology-based querying," in *Proceedings of the 4th International Conference on Flexible Query-Answering Systems (FQAS'00)*, Warsaw, Poland, October 2000, pp. 15–26.
[29] P. Buche, O. Haemmerlé, and R. Thomopoulos, "Integration of heterogeneous, imprecise and incomplete data: an application to the microbiological risk assessment," in *Proceedings of the 14th International Symposium on Methodologies for Intelligent Systems (ISMIS'2003), Lecture Notes in Artificial Intelligence*. Maebashi, Japan: Springer, October 2003, pp. 98–107.
[30] R. Thomopoulos, P. Buche, and O. Haemmerlé, "Representation of weakly structured imprecise data for fuzzy querying," *Fuzzy Sets and Systems*, vol. 140, pp. 111–128, 2003.
[31] P. Buche, J. Dibie-Barthélemy, O. Haemmerlé, and G. Hignette, "Fuzzy semantic tagging and flexible querying of XML documents extracted from the web," *Journal of Intelligent Information Systems*, vol. 26, no. 1, pp. 25–40, 2005.

Rallou Thomopoulos defended her Ph.D thesis in computer science in 2003. She has been a researcher in knowledge representation at the INRA national research institute since 2004, and associate researcher of the LIRMM computer science laboratory of Montpellier since 2006. She works on the integration of ontology- and data-representation formalisms, the expression of gradual information and viewpoints in ontologies, and more specifically in the framework of the conceptual graph model.



Patrice Buche received the Ph.D degree in computer science from the University of Rennes (France) in 1990. He has been a research engineer at the Mathematical and Computer Science department of INRA (Institut National de la Recherche Agronomique) since 2002 and an assistant professor at INA P-G (Institut National Agronomique Paris-Grignon) since 1992. His research works mainly concern data integration and fuzzy querying in structured and weakly-structured databases. Dr Buche has published papers in these areas in several international conferences and journals.

Ollivier Haemmerlé defended his phD concerning the Conceptual Graph model in 1995. He has been a professor in Toulouse 2 University and a member of the GRIMM-Isycom laboratory since September 2005. Formerly, he was an assistant professor in Institut National Agronomique Paris-Grignon from 1996 to 2005 and a member of the Laboratoire de Recherche en Informatique of the French University Paris Sud from 2003 to 2005. He works on the representation of microbiological data by means of Conceptual Graphs and on the validation of such knowledge. He also works on the semantic Web, particularly on the representation and querying of semi-structured data represented in XML and in the integration of heterogeneous data.