An approach to solve division-like queries in fuzzy object databases

Nicolás Marín\textsuperscript{a,}\textsuperscript{*}, Carlos Molina\textsuperscript{b}, Olga Pons\textsuperscript{a}, M. Amparo Vila\textsuperscript{a}

\textsuperscript{a} Department of Computer Science and A.I., University of Granada, 18071 Granada, Spain
\textsuperscript{b} Department of Informatics, University of Jaén, Campus Las Lagunillas, A3, 23071, Jaén, Spain

Available online 29 December 2011

Abstract

The division operator is well known in the context of fuzzy relational databases. Fuzzy approaches to this operator can be found in the literature to solve flexible queries: these approaches try to compute to which extent each element of a candidate set is connected with a given (fuzzy) set of elements. This kind of queries is also interesting in the framework of fuzzy object databases, where the management of complex objects instead of plain tuples causes additional difficulties. The presence of fuzzily described objects in the database makes necessary to use suitable operators that take into account the resemblance that governs the comparison in the underlying reference universe. In this paper, we propose a method to solve this type of queries founded on the use of fuzzy inclusion operators. We study two different alternatives for the way resemblance is considered and we also analyze the role of cardinality in the process.

© 2012 Elsevier B.V. All rights reserved.

Keywords: Division; Flexible queries; Fuzzy object databases; Inclusion

1. Introduction

The management of data affected by vagueness has been deeply studied by the research community in order to improve the capability of data handling in conventional databases and increase the range of problems that can be modeled. In this context, the use of Fuzzy Sets Theory to deal with this vagueness has produced as result the launch of Fuzzy Database Systems. At the present time, we can state that they have proved to be an adequate tool for the storage and management of data when they are affected by imperfections of different kinds.

One of the most important data paradigms in both the programming and the databases world is the Object-Oriented Data Model. Modeling the reality which is behind many software problems as a set of objects grouped around classes has proved to be a beneficial approach for many developers, particularly when dealing with complex and dynamic problems. In the database area, founded on many concepts of this data model, object-oriented and object-relational database management systems allow the representation of schema when complex relationships make it difficult the use of Codd’s relational model. The object-oriented data model is more powerful from a modeling point of view thanks to important features such as inheritance and encapsulation. In fact, most of commercial database management systems, though they are not purely object-oriented, are indeed offering an extended relational model to cope with object-orientation.
As it was the case with the relational data model in the past, many researchers have also tried to improve object orientation with the help of fuzzy concepts. Advanced semantic data models can be considered as the origin of the study of fuzziness in object-oriented models. Rossazza et al. [1] introduced a hierarchical model of fuzzy classes and George et al. [2] began to use similarity relationships in order to model attribute value imperfection. The work of George et al. has been completed by Yazici et al. [3], with IFOOD as result, an intelligent fuzzy object-oriented data model. At the end of nineties, Fuzzy Object-Oriented Database Models became an independent research field within the database research area: Bordogna and Passi [4] introduced an extended graphical notation to represent fuzzy object-oriented information, while Van Gyseghem and de Caluwe [5] developed the UFO model, one of the most complete proposals that can be found in the literature. In the last years, in [6] a framework is presented which allows to handle imprecision in object models, for example, in the description of complex objects. The framework is founded on the use of fuzzy types [7,8] and fuzzily described objects. In the context of object-oriented programming, fuzzy proposals can also be found in the literature for platforms such as Java [9] and .NET [10]. More recently, the object-relational data model has also been extended so that fuzzy data can be stored and managed (see for example [11,12]). The interested reader can find significant surveys of works related to this topic in [13–15].

An important part of the efforts in the research work of this fuzzy database community has been devoted to the management of fuzzy queries. Fuzzy queries have been specially studied in the relational context. This way, the most common operators of the relational algebra have been deeply studied. Though relational division is not part of the basic Codd’s relational algebra operators and it is not directly implemented in SQL, this operator has been profusely investigated in the context of fuzzy relational databases [16–23].

The analysis of this kind of queries is not so profuse in object databases, probably motivated by the lack of a standard set of algebraic operations [24]. In any case, queries with a similar semantics to the one of the division operator can also appear in (fuzzy) object databases: there are situations where we need to compute to which extent each element of a candidate set (dividend) is connected with a given (fuzzy) set of elements (divisor). In the case of fuzzy object data models, the problem is more complex because the elements can be objects; furthermore, they can be fuzzily described objects. This imprecision in the description of objects generates a problem with equality in the universe of objects which is below the queries we want to solve. That is, instead of looking for the connection of the candidate element with a rigid divisor, we can look for the connection with a set of objects similar enough to the given divisor. As we will see, to solve this type of queries in a fuzzy object database, we need to study suitable operators that appropriately extend the inclusion notion so that resemblance can be appropriately taken into account and object complexity can be suitably managed. Cardinality issues are also of crucial importance in the process since, for example, they can be used to relax the universal quantifier implicit in the classical division. In this paper, we will also analyze them in depth.

A crucial objective of our approach is that it can be used through simple SQL99-compliant sentences. In this paper we also present how the proposed theoretical framework can be implemented inside a fuzzy object database system like pg4db [12].

The paper is organized as follows: Section 2 briefly introduces the fuzzy object-oriented data model that is the basis of our proposal and describes the kind of queries we focus on; Section 3 analyzes the problem of considering the resemblance when computing the inclusion associated to the query while in Section 4, we analyze the role that the cardinality plays in the resolution of the query; examples in a proof of concept implementation of the proposed operators are offered in Section 5 together with some implementation issues. Finally, some conclusions are outlined in Section 6.

2. The model and the problem

This section is devoted to present the problem discussed in this paper and its context. First subsection briefly presents the main characteristics of our fuzzy object model [9,6]. Then, in the second subsection, we describe the kind of queries we focus on in this work.
2.1. A fuzzy object-oriented model

In any object model, the state of an object is equated with a set of attribute values according to its class description. The approach described in [9,6] considers different types of attribute values in order to give support to the representation of fuzzily described objects. Together with the precise values, the objects, and the crisp collections, this fuzzy model permits to use more powerful attribute values, as imprecise labels and fuzzy collections.

Fig. 1 shows a description of the different types of attribute values that can be used to set up an object in the model. That is, the object state can be composed by:

- **Precise values**: This category of values involves all the classical basic classes that usually appear in an object-oriented data model (e.g. numerical classes, string classes, etc.). Values in these domains are easily represented and compared using conventional built-in data types and the classical set of relational operators.
- **Imprecise values**: The case of imprecise (atomic) values is a bit more complex. In many cases, linguistic labels [25–27] are associated to this kind of values.
- **Objects**: The attribute value may be a reference to another object (constituting what is called a complex object).
- **Collections**: The attribute may be conformed by a set of values or, even, by a set of objects. Imprecision in this kind of attributes may appear at two levels:
  - The set may be fuzzy. The semantics of membership degrees depends on the problem, but, in collections, the fuzzy set is considered to be conjunctive. For example, the set of languages a person can speak may be fuzzy if we take into account the degree of expertise for every language. In this case, we have a fuzzy collection of precise objects.
  - The elements of the set may be fuzzy values or, in general, fuzzily described objects. For example, a collection of people, where every person is imprecisely described (e.g., using linguistic labels to describe the height and the weight). In this case, we have a precise collection of imprecise objects.

An example with both kinds of imprecision may be the collection of friends of a given person: if we use a degree to measure the friendship relation, then the collection is a fuzzy set. If objects of the class Friend are described by means of fuzzy attribute values, then the elements of the collection are also imprecise themselves.
2. Generalization of equality

We do not only need a representation of fuzzily described objects in our model but also a way to manage this kind of objects. This includes a basic capability to compare the state of two objects belonging to a given class.

With fuzzily described objects like the ones presented in the previous paragraphs, this comparison has to be performed by a process which consists of two steps: computing degrees of resemblance for pairs of attribute values and, then, aggregating these resemblance degrees to obtain a general degree of object resemblance.

Consider a given class $C$ in our fuzzy object model. The class description has a type $T_C$ whose structural component $\text{Str}_C$ is an attribute set $\{x_1, x_2, \ldots, x_n\}$. If $o_1$ and $o_2$ are two objects belonging to class $C$, in order to compare them, we have to aggregate the resemblance degrees obtained for every pair $(o_1.x_i, o_2.x_i)$.

In this process, $S_{x_i}(o_1, o_2)$ stands for the resemblance degree observed between the $x_i$ attribute values for objects $o_1$ and $o_2$, and $S(o_1, o_2)$ is the aggregated resemblance degree we want to calculate.

In the computation, not all attributes need to have the same importance: each attribute $x_i$ has a weight $p_{x_i} \in [0, 1]$ associated with it that represents its relative importance in the final decision.

Fig. 2 summarizes this comparison process: in order to compare objects $o_1$ and $o_2$, we first compare their attribute values and obtain particular resemblance degrees $S_{x_i}(o_1, o_2)$. Then, we aggregate them to obtain a global resemblance opinion $S(o_1, o_2)$ according to the attribute importance established in the class.

As can be observed, the comparison of two objects involves a recursive procedure whose base case is found in the basic domains. For the sake of space, we omit here the operators for base cases and the analysis of the recursive aggregation procedure used in our model. Interested readers can find a detailed description in [28,10].

2.3. Division queries in fuzzy object databases

In this section, we first focus on the division in fuzzy relational databases and, then, we describe the division-like queries that we want to solve in a fuzzy object-oriented context.

Suppose that we have two relations $t_a$ and $t_b$ whose schema are, respectively, $T_A(X, Y)$ and $T_B(Y')$, where $Y$ and $Y'$ are compatible attributes. The relational division $t_a \div t_b$ delivers a new relation $t_d$ with schema $T_D(X)$ and with the following set of tuples:

$$\{a | a \in \text{domain}(X) \land (\forall b, (b \in t_b) \Rightarrow (< a, b > \in t_a))\}$$

(1)
Assume now that \( t_a \) and \( t_b \) are fuzzy relations, i.e., that their tuples are weighted by a number between 0 and 1. That is, each tuple of \( t_a \) is affected by a membership degree, \( \mu_{t_a}(<a,b>) \in [0,1] \) and, similarly, each tuple of \( t_b \) has its corresponding \( \mu_{t_b}(<b>) \in [0,1] \).

Each \( a \in \text{domain}(X) \) belongs to the division result \( t_d \) with a membership degree that can be obtained as follows [18]:

\[
\mu_{t_d}(a) = \inf_b \{ I(\mu_{t_b}(b), \mu_{t_a}(a,b)) \}
\]

where \( I \) stands for a suitable fuzzy implication according to the desired properties.

The previous paragraphs briefly summarize the division problem in fuzzy relational databases. However, in the case of an object model, we have to consider (in the more general case) that attribute values are objects. That is, in \( t_a \), each element \( a_i \) of the candidate set is related with a set of objects \( \{ \text{object}_{i,1}, \ldots, \text{object}_{i,n_i} \} \). This relation is fuzzy, and \( \mu_{t_a}(a_i, \text{object}_j) \) is the degree of relationship between \( a_i \) and \( \text{object}_j \).

Similarly, the divisor \( t_b \) is a fuzzy set of objects

\[
\mu_{t_b}(\text{object}_{b,1})/\text{object}_{b,1} + \cdots + \mu_{t_b}(\text{object}_{b,k})/\text{object}_{b,k}
\]

The result \( t_d \) is a fuzzy set defined over the candidate set, where \( \mu_{t_d}(a_i) \) indicates to which extent element \( a_i \) is related in \( t_a \) with all the elements of the divisor.

Fig. 3 graphically depicts the stated problem. As in an object-oriented model attribute values do not necessarily have to be atomic, \( t_a \) (and, similarly \( t_b \)) might have the schema shown in Fig. 4.

If, for any candidate element \( a_i \), we define a fuzzy set of objects \( A_i \) with the following membership function:

\[
\mu_{A_i}(\text{object}_j) = \mu_{t_a}(a_i, \text{object}_j)
\]
in order to compute $\mu_{t_b}(a_i)$ we have to compute to which extent $t_b$ is included in $A_i$.

That is, whatever the schema version of dividend and divisor tables are, the solution for division queries entails the resolution of the inclusion problem described in Fig. 5.

As an example, consider an object database built by a certain promoter with information about construction companies. In the example, attribute X may refer to the code of a construction company and attribute Y may contain the list of construction machinery they have. The promoter weights the relationship of each pair company-machinery according to the supposed expertise of the company in relation with the machinery. In the database, there is a class construction machinery and objects of this class can be (fuzzily) compared according to its description from the construction point of view.

Imagine that the promoter elaborates a list of machineries that are needed to carry out a certain project, also weighted according to the demanded expertise level. In this database, an example query of the kind we focus on in this paper could be to find those companies which can work with all the listed machinery.

Next sections are devoted to analyze this inclusion problem.

3. Resemblance and inclusion

According to the previous paragraphs, in order to solve division queries in object models, we need to suitably compute the above mentioned inclusion. That is, to compute the inclusion between any couple of fuzzy sets of objects (like in Fig. 5), we need to generalize the fuzzy inclusion operators in use in the division, taking into account that the objects of the set may be fuzzily described.

Several proposals for the calculus of the inclusion degree between fuzzy sets can be found in the literature. In [1] the inclusion degree between two fuzzy sets $B$ and $A$ is calculated as follows:

$$N(A|B) = \min_{u \in U} \{I(\mu_B(u), \mu_A(u))\}$$

where $I$ stands for a fuzzy implication operator, and $\mu_X$ is the membership function that describes the fuzzy set $X$. This degree coincides with the fuzzy division operator described in (2). The implication operator can be chosen in accordance to the properties we want the inclusion degree to fulfil [18,19].

Nevertheless, independently of the chosen implication operator, this formulation supposes that both $A$ and $B$ are defined over a reference universe $U$ made up of precise elements, where the classical equality is the basis of the comparisons. That is, the implication operator compares to what extent the presence of an element of the universe $U$ in $B$ forces its presence in $A$ (i.e. we compare the membership degrees of the same object to both sets).

However, in the context of fuzzy object databases, it frequently happens that the elements of the universe $U$ are imprecise objects among which classical equality cannot be applied (as we commented in Section 2). Instead, a resemblance relationship must be used. That is, for a given element in the set $B$, it is not straightforward which element of $A$ has to be taken in order to compare the membership degrees.

In our context of object databases, the objects of the two sets of Fig. 5 may be defined with imprecision and two apparently different objects (from the identity point of view) may be similar (from the value equality point of view). In the example of the construction database cited at the end of the previous section, two construction machineries can be matched according to their properties from the construction point of view, though their oids (object identifiers) are different.
In the first approach, for each element into account resemblance among the involved objects, as well as the mentioned implication, is of interest.

In this paper, we consider two ways of doing that:

- In the first approach, for each element \(x\) of \(B\), we look for an element \(y\) of the universe \(U\) that fulfills two constraints [28]:
  - It sufficiently belongs to \(A\) (according to the mentioned fuzzy implication).
  - It is similar enough to \(x\).

  That is, we consider the following restriction:

  \[
  (\mu_B(x) \Rightarrow \mu_A(y)) \land \mu_S(x, y)
  \]

  \( (5) \)

- In the second approach, as suggested in [29], we can expand the dividend according to the resemblance relation, so that we consider that elements similar to the original elements of \(A\) are also present in the expanded dividend. The membership to this enlarged version of the dividend is obtained by composing membership to \(A\) with the resemblance relation. Thus, in this case, we consider the following restriction:

  \[
  \mu_B(x) \Rightarrow (\mu_A(y) \land \mu_S(x, y))
  \]

  \( (6) \)

In summary, in the first approach, for each element \(x \in \text{support}(B)\) we look for an element \(y \in \text{support}(A)\) similar to \(x\) that fulfills the implication on their original memberships. In the second approach, we try to increase the membership of \(x\) to \(A\) according to the memberships of similar elements that are present in \(A\) \((y \in \text{support}(A))\). As we will see in the following subsections, both approaches lead to suitable flexible inclusion operators though the second one usually delivers more optimistic results when usual pairs of t-norm and implications are used.

### 3.1. Resemblance as a bound of the implication

In order to deal with resemblance when computing the inclusion of two fuzzy sets of fuzzily described objects, (4) can be generalized as follows [28]:

**Definition 1 (Resemblance driven inclusion degree).** Let \(A\) and \(B\) be two fuzzy sets defined over a finite reference universe \(U\), \(S\) be a resemblance relation defined over the elements of \(U\), \(\otimes\) be a t-norm and, \(I\) an implication defined according to \(\otimes\). The inclusion degree of \(B\) in \(A\) driven by the resemblance relation \(S\) is calculated as follows:

\[
\Theta_S(A|B) = \min_{x \in U} \max_{y \in U} \theta_{B,A,S}(x, y)
\]

where

\[
\theta_{B,A,S}(x, y) = \otimes(I(\mu_B(x), \mu_A(y)), \mu_S(x, y))
\]

(7)

In (4), the inclusion degree is calculated taking into account the element of \(U\) that fulfills in a lower degree the implication condition between the membership degrees of this element to both sets. In (7), since the membership degrees of similar (but not necessarily equal from the identity point of view) elements can be compared, we restrict the implication using the resemblance degree of the two elements. In summary, for each element that belongs with a certain degree to the set \(B\), we look for a quite similar object in \(U\) that belongs to the set \(A\) with an equal or higher degree.

Consider the example of Fig. 6. In this example, \(\{\text{object}_1, \text{object}_2, \text{object}_3, \text{object}_4\}\) is the reference universe \(U\) over which the resemblance relation \(S\) of Table 1 is defined.

In this situation: \(\Theta_S(A|B)=\min \{ \max \{ \otimes(I(\mu_B(\text{object}_1), \mu_A(\text{object}_1)), \mu_S(\text{object}_1, \text{object}_1)\), \ldots, \otimes(I(\mu_B(\text{object}_1), \mu_A(\text{object}_4)), \mu_S(\text{object}_1, \text{object}_4)) \}, \ldots, \max \{ \otimes(I(\mu_B(\text{object}_4), \mu_A(\text{object}_1)), \mu_S(\text{object}_4, \text{object}_1)\}, \ldots, \otimes(I(\mu_B(\text{object}_4), \mu_A(\text{object}_4)), \mu_S(\text{object}_4, \text{object}_4)) \}\}.

If we use the minimum as t-norm and the Gödel’s residuated implication operator (9), then

\[
I(x, y) = \begin{cases} 
1 & \text{if } x \preceq y \\
y & \text{otherwise}
\end{cases}
\]

(9)

\[
\Theta_S(A|B)= \min \{ \max \{0,0,0,0\}, \max \{0,6,1,0\}, \max \{0,0,1\}, \max \{0,0,1\}\} = \min\{0,6,1,1,1\}=0.6.
\]
Proposition 1. Let $X_1$ and $X_2$ be two fuzzy sets defined over a finite reference universe $\mathcal{U}$ over which a resemblance relation $S$ is defined. Let $S$ be the classic equality relation. Then, $\Theta_\circ(X_2|X_1) = N(X_2|X_1)$.

Proof. $\Theta_\circ(X_2|X_1) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \theta_{X_1,X_2,\circ}(x, y)$. Since $\theta_{X_1,X_2,\circ}(x, y) = \ominus(I(\mu_{X_1}(x), \mu_{X_2}(y)), \mu_{\circ}(x, y))$ and taking into account that $\forall x \neq y$, $\mu_{\circ}(x, y) = 0$ then $\forall x \neq y$, $\theta_{X_1,X_2,\circ}(x, y) = \ominus(I(\mu_{X_1}(x), \mu_{X_2}(y)), 0) = 0$ which yields $\forall x \in \mathcal{U}$, $\max_{y \in \mathcal{U}} \theta_{X_1,X_2,\circ}(x, y) = \theta_{X_1,X_2,\circ}(x, x) = \ominus(I(\mu_{X_1}(x), \mu_{X_2}(x)), 1) = I(\mu_{X_1}(x), \mu_{X_2}(x))$ and thus $\Theta_\circ(X_2|X_1) = \min_{x \in \mathcal{U}} I(\mu_{X_1}(x), \mu_{X_2}(x)) = N(X_2|X_1)$. \qed

- On the other hand, the resemblance driven inclusion degree ($\Theta$) is the inclusion operator of Eq. (4) when classic equality is used.

Proposition 2. Let $X_1$, $X_2$, and $X_3$ be three fuzzy sets defined over a finite reference universe $\mathcal{U}$ over which a resemblance relation $S$ is defined. Then, $X_1 \subseteq X_2 \Rightarrow \Theta_S(X_3|X_1) \geq \Theta_S(X_3|X_2)$.

Proof. $X_1 \subseteq X_2 \Rightarrow \forall x \in \mathcal{U}, \mu_{X_1}(x) \leq \mu_{X_2}(x)$. That is, $\forall x \in \mathcal{U}, \mu_{X_1}(x) + \lambda = \mu_{X_2}(x)$, with $\lambda \geq 0$. Also $\text{Support}(X_1) \subseteq \text{Support}(X_2)$. By the properties of implications, $\Theta_S(X_3|X_1) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \theta_{X_1,X_3,S}(x, y) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \ominus(I(\mu_{X_1}(x), \mu_{X_3}(y)), \mu_S(x, y)) \geq \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \ominus(I((\mu_{X_1}(x) + \lambda), \mu_{X_3}(y)), \mu_S(x, y))$, $\forall \lambda \geq 0$. Particularly, $\Theta_S(X_3|X_1) \geq \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \ominus(I(\mu_{X_2}(x), \mu_{X_3}(y)), \mu_S(x, y)) = \Theta_S(X_3|X_2)$. \qed

- On the other hand, the resemblance driven inclusion degree ($\Theta$) is monotonic with respect to the inclusion relationship.

Proposition 3. Let $X_1$, $X_2$, and $X_3$ be three fuzzy sets defined over a finite reference universe $\mathcal{U}$ over which a resemblance relation $S$ is defined. Then, $X_1 \subseteq X_2 \Rightarrow \Theta_S(X_2|X_3) \geq \Theta_S(X_1|X_3)$.

Proof. $X_1 \subseteq X_2 \Rightarrow \forall x \in \mathcal{U}, \mu_{X_1}(x) \leq \mu_{X_2}(x)$. That is, $\forall x \in \mathcal{U}, \mu_{X_1}(x) + \lambda = \mu_{X_2}(x)$, with $\lambda \geq 0$. Also $\text{Support}(X_1) \subseteq \text{Support}(X_2)$. By the properties of t-norms and implications, $\Theta_S(X_2|X_3) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \theta_{X_2,X_3,S}(x, y) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \ominus(I((\mu_{X_1}(x), \mu_{X_2}(y)), \mu_S(x, y)) \geq \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \ominus(I((\mu_{X_2}(x), \mu_{X_2}(y) - \lambda), \mu_S(x, y))$, $\forall \lambda \geq 0$. Particularly, $\Theta_S(X_2|X_3) \geq \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \ominus(I(\mu_{X_1}(x), \mu_{X_2}(y)), \mu_S(x, y)) = \Theta_S(X_1|X_3)$. \qed

Fig. 7(a) depicts the intermediate results obtained when computing the value of $\Theta_S(A|B)$ in the example.

There are three properties of the resemblance driven inclusion degree ($\Theta$) that are of special interest in relation with the semantics of division queries [28]:

- On the one hand, the resemblance driven inclusion degree ($\Theta$) is the inclusion operator of Eq. (4) when classic equality is used.
3.2. Using resemblance to modulate the membership to A

As an alternative to the strategy used in the previous subsection, we can consider that the resemblance only restricts the second operand of the implication. The effect of this second version of flexible inclusion is to enlarge the set \( A \) into a superset \( A' \) obtained by composing \( A \) with the resemblance relation, as suggested in [29].

The idea is to expand \( A \) so that those objects of \( U \) similar enough to the initial objects of \( A \), are added to \( A \). This resemblance based tolerant inclusion can be defined as follows:

**Definition 2 (Resemblance based tolerant inclusion degree).** Let \( A \) and \( B \) be two fuzzy sets defined over a finite reference universe \( U \), \( S \) be a resemblance relation defined over the elements of \( U \), \( \otimes \) be a t-norm, and \( I \) an implication defined according to \( \otimes \). The tolerant inclusion degree of \( B \) in \( A \) according to the resemblance relation \( S \) is calculated as follows:

\[
\Psi_S(A|B) = \min_{x \in U} \max_{y \in U} \psi_{B,A,S}(x, y)
\]

where

\[
\psi_{B,A,S}(x, y) = I(\mu_B(x), \otimes(\mu_A(y), \mu_S(x, y)))
\]

According to the example of Fig. 6:

\[
\Psi_S(A|B) = \min \{ \max I(\mu_B(object_1), \otimes(\mu_A(object_1), \mu_S(object_1, object_1))), \ldots, I(\mu_B(object_1), \otimes(\mu_A(object_4), \mu_S(object_1, object_4))), \ldots, \max I(\mu_B(object_4)) \}
\]
\( \otimes(\mu_A(object_4), \mu_S(object_4, object_1)) \), ..., \( I(\mu_B(object_4), \otimes(\mu_A(object_4), \mu_S(object_4, object_4))) \).

Again, if we use the minimum as t-norm and the Gödel's residuated implication operator \( \Theta \), then:

\[
\Psi_S(A|B) = \min\{\max\{0, 1, 0, 0\}, \max\{1, 1, 1\}, \max\{0, 0, 1, 0\}, \max\{1, 1, 1\}\} = \min\{1, 1, 1\} = 1.
\]

Fig. 7(b) depicts the intermediate results obtained when computing the value of \( \Theta_S(A|B) \) in the example.

This \( \Psi \) operator has counterpart properties to the ones described for \( \Theta [30] \):

**Proposition 4.** Let \( X_1 \) and \( X_2 \) be two fuzzy sets defined over a finite reference universe \( \mathcal{U} \) over which a resemblance relation \( S \) is defined. Let \( S \) be the classic equality relation. Then, \( \Psi = \Psi(X_2|X_1) = N(X_2|X_1) \).

**Proof.** \( \Psi = \Psi(X_2|X_1) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \Psi_{X_1,X_2,=}(x, y) = I(\mu_{X_1}(x), \otimes(\mu_{X_2}(y), \mu_S(y, x))) \) and taking into account that \( \forall x \neq y, \mu_{X_1,X_2,=}(x, y) = I(\mu_{X_1}(x), \otimes(\mu_{X_2}(y), 0)) = 0 \) which yields \( \forall x \in \mathcal{U}, \max_{y \in \mathcal{U}} \Psi_{X_1,X_2,=}(x, y) = \Psi_{X_1,X_2,=}(x, x) = I(\mu_{X_1}(x), \otimes(\mu_{X_2}(x), 1)) = I(\mu_{X_1}(x), \mu_{X_2}(x)) \) and thus \( \Psi = \Psi(X_2|X_1) = \min_{x \in \mathcal{U}} \{I(\mu_{X_1}(x), \mu_{X_2}(x))\} = N(X_2|X_1) \).

**Proposition 5.** Let \( X_1, X_2, \) and \( X_3 \) be three fuzzy sets defined over a finite reference universe \( \mathcal{U} \) over which a resemblance relation \( S \) is defined. Then, \( X_1 \subseteq X_2 \Rightarrow \Psi_S(X_3|X_1) \geq \Psi_S(X_3|X_2) \).

**Proof.** \( X_1 \subseteq X_2 \Rightarrow \forall x \in \mathcal{U}, \mu_{X_1}(x) \leq \mu_{X_2}(x) \). That is, \( \forall x \in \mathcal{U}, \mu_{X_1}(x) + \lambda = \mu_{X_2}(x) \), with \( \lambda \geq 0 \). Also \( \text{Support}(X_1) \subseteq \text{Support}(X_2) \). By the properties of implications, \( \Psi_S(X_3|X_1) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \Psi_{X_1,X_2,S}(x, y) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} I(\mu_{X_1}(x), \otimes(\mu_{X_2}(y), \mu_S(y, x))) \geq \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} I((\mu_{X_1}(x) + \lambda, \otimes(\mu_{X_1}(y), \mu_S(x, y)))) \forall \lambda \geq 0 \). Particularly, \( \Psi_S(X_3|X_1) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} I((\mu_{X_1}(x), \otimes(\mu_{X_2}(y), \mu_S(x, y)))) = \Psi_S(X_3|X_2) \).

**Proposition 6.** Let \( X_1, X_2, \) and \( X_3 \) be three fuzzy sets defined over a finite reference universe \( \mathcal{U} \) over which a resemblance relation \( S \) is defined. Then, \( X_1 \subseteq X_2 \Rightarrow \Psi_S(X_3|X_2) \geq \Psi_S(X_3|X_3) \).

**Proof.** \( X_1 \subseteq X_2 \Rightarrow \forall x \in \mathcal{U}, \mu_{X_1}(x) \leq \mu_{X_2}(x) \). That is, \( \forall x \in \mathcal{U}, \mu_{X_1}(x) + \lambda = \mu_{X_2}(x) \), with \( \lambda \geq 0 \). Also \( \text{Support}(X_1) \subseteq \text{Support}(X_2) \). By the properties of t-norms and implications, \( \Psi_S(X_2|X_3) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} \Psi_{X_1,X_2,S}(x, y) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} I(\mu_{X_1}(x), \otimes(\mu_{X_2}(y), \mu_S(x, y))) \geq \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} I((\mu_{X_1}(x), \otimes(\mu_{X_2}(y), \mu_S(x, y)))) \forall \lambda \geq 0 \). Particularly, \( \Psi_S(X_2|X_3) = \min_{x \in \mathcal{U}} \max_{y \in \mathcal{U}} I((\mu_{X_1}(x), \otimes(\mu_{X_2}(y), \mu_S(x, y)))) = \Psi_S(X_1|X_3) \).

3.3. Comparison of the two proposed methods

We have introduced in the previous subsections two different ways of solving the inclusion problem we deal with. Now, we are going to analyze more in depth the behavior of both \( \Theta \) and \( \Psi \) operators.

As we have already shown, resemblance is acting as a bound of the implication in operator \( \Theta \) while, in operator \( \Psi \), resemblance acts on the membership to \( A \) of the considered object. Both operators are appropriate to face the inclusion problem; however, the results they offer differ depending on the data they work with and the t-norm and implication used to build them. In this subsection, we will see that the use of \( \Psi \) leads to more optimistic results than the use of \( \Theta \) when using usual well-known couples of operators for the t-norm and the implication under these operators.

3.3.1. Using minimum and Gödel

When computing the tolerant inclusion, in the calculation of \( \Psi_{B,A,S}(x, y) \), the membership degree to \( B \) of a given object \( x \) (which is at the left part of the implication), bounds the needed resemblance between \( x \) and \( y \) (that restrict the membership of \( y \) to \( A \) in the right part of the implication). This fact could produce a big change in \( \Psi \) in the presence of a small change in the resemblance relation and permits to this operator to deliver higher results than \( \Theta \) when using usual pairs t-norm/implication.

Consider again the example of Fig. 6 and Table 1. For this example, using minimum and Gödel, \( \Theta_S(A|B) = 0.6 \) and \( \Psi_S(A|B) = 1 \). If we change the resemblance relation of Table 1 so that \( \mu_S(object_1, object_2) = 0.5 \) and maintain the rest of data, then \( \Theta_S(A|B) = 0.5 \) and \( \Psi_S(A|B) = 0.5 \) (Fig. 8(a) and (b) depicts the intermediate results obtained when computing the value of \( \Theta_S(A|B) \) and \( \Psi_S(A|B) \)). As can be seen, a little change in \( \mu_S(object_1, object_2) \) considerably reduces the value of \( \Psi_S(A|B) \) while \( \Theta_S(A|B) \) slightly decreases.
In any case, using the minimum as t-norm and Gödel as implication operator, we can prove that

\[\text{If, on the contrary, we consider that } \mu_S(\text{object}_1, \text{object}_2) = 0.8, \text{ then we have that } \Theta_S(A|B)=0.8 \text{ and } \Psi_S(A|B)=1.\]

In any case, using the minimum as t-norm and Gödel as implication operator, we can prove that \( \Psi_S(A|B) \geq \Theta_S(A|B), \forall A, B. \) To do that, it is enough to prove that \( \forall \mu_A(x), \mu_B(x), \text{ and } \mu_S(x, y) \)

\[\otimes(I(\mu_B(x), \mu_A(y)), \mu_S(x, y)) \leq I(\mu_B(x), \otimes(\mu_A(y), \mu_S(x, y)))\]

\textbf{Proof.} Let \( \alpha, \beta, \gamma \in [0, 1]. \) Using the minimum as t-norm and Gödel as implication operator, we can prove that \( \otimes(I(\beta, x), \gamma) \leq I(\beta, \otimes(x, \gamma)) \) by means of the following case analysis:

- **Case 0:** \( \beta = 0. \) Since \( I(\beta, \min(x, \gamma)) = 1, \) then \( \min(I(\beta, x), \gamma) = I(\beta, min(x, \gamma)). \)
- **Case 1:** \( \gamma \leq \beta \leq x, \beta \neq 0. \) Since \( \min(I(\beta, x), \gamma) = min(1, \gamma) = \gamma \) and \( I(\beta, \min(x, \gamma)) = I(\beta, \gamma) = \gamma, \) then \( \min(I(\beta, x), \gamma) \leq I(\beta, min(x, \gamma)). \)
- **Case 2:** \( \gamma \leq \alpha \leq \beta, \beta \neq 0. \) Since \( \min(I(\beta, x), \gamma) = min(1, \gamma) = \gamma \) and \( I(\beta, \min(x, \gamma)) = I(\beta, \gamma) = \gamma, \) then \( \min(I(\beta, x), \gamma) \leq I(\beta, min(x, \gamma)). \)
- **Case 3:** \( \beta \leq \gamma \leq \alpha, \beta \neq 0. \) Since \( \min(I(\beta, x), \gamma) = min(1, \gamma) = \gamma \) and \( I(\beta, \min(x, \gamma)) = I(\beta, \gamma) = 1, \) then \( \min(I(\beta, x), \gamma) \leq I(\beta, min(x, \gamma)). \)
- **Case 4:** \( \beta \leq \alpha \leq \gamma, \beta \neq 0. \) Since \( \min(I(\beta, x), \gamma) = min(1, \gamma) = \gamma \) and \( I(\beta, \min(x, \gamma)) = I(\beta, \gamma) = 1, \) then \( \min(I(\beta, x), \gamma) \leq I(\beta, min(x, \gamma)). \)
- **Case 5:** \( \alpha \leq \gamma \leq \beta, \beta \neq 0. \) Since \( \min(I(\beta, x), \gamma) = min(1, \gamma) = \gamma \) and \( I(\beta, \min(x, \gamma)) = I(\beta, \gamma) = \gamma, \) then \( \min(I(\beta, x), \gamma) \leq I(\beta, min(x, \gamma)). \)
• Case 6: \( \alpha \leq \beta \leq \gamma, \beta \neq 0 \). Since \( \min(I(\beta, \alpha), \gamma) = \min(\alpha, \gamma) \leq \alpha \) and \( I(\beta, \min(\alpha, \gamma)) = I(\beta, \alpha) = \alpha \), then \( \min(I(\beta, \alpha), \gamma) \leq I(\beta, \min(\alpha, \gamma)) \).

3.3.2. The case of product and Goguen

In any case, the behavior of \( \Theta \) and \( \Psi \) depends on the used operators. We can observe this dependence if we use product as t-norm and Goguen (see Eq. (12)) as implication. Though \( \Psi \) is still more tolerant than \( \Theta \), the effect of resemblance changes is different

\[
I(x, y) = \begin{cases} 
1 & \text{if } x \leq y \\
y/x & \text{otherwise}
\end{cases}
\] (12)

With this t-norm and this implication, the results for the same example are as follows:

• With the original resemblance relation (Table 1): \( \Theta_S(A|B) = 0.6 \) and \( \Psi_S(A|B) = 0.7 \).
• With the second resemblance relation \( (\mu_S(\text{object}_1, \text{object}_2) = 0.5) : \Theta_S(A|B) = 0.5 \) and \( \Psi_S(A|B) = 0.58 \).
• With the last modification in resemblance \( (\mu_S(\text{object}_1, \text{object}_2) = 0.8) : \Theta_S(A|B) = 0.8 \) and \( \Psi_S(A|B) = 0.93 \).

With this configuration of t-norm and implication, the effect of changing similarity is not so big, though we also observe that \( \Theta_S(A|B) \leq \Psi_S(A|B) \).

In fact, this inequality can be proved \( \forall A, B \).

Proof. Let \( \alpha, \beta, \gamma \in [0, 1] \). Using the product as t-norm and Goguen as implication operator, we can proof that \( \otimes(I(\beta, \alpha), \gamma) \leq I(\beta, \otimes(\alpha, \gamma)) \) by means of the following case analysis:

• Case 0: \( \beta = 0 \). Since \( I(\beta, \alpha) \ast \gamma = \gamma \) and \( I(\beta, \alpha \ast \gamma) = 1 \), then \( I(\beta, \alpha) \ast \gamma \leq I(\beta, \alpha \ast \gamma) \).
• Case 1: \( \gamma \leq \beta \leq \alpha, \beta \neq 0 \). Since \( I(\beta, \alpha) \ast \gamma = \gamma \) and \( I(\beta, \alpha \ast \gamma) \geq \min(\alpha \ast \gamma / \beta, 1) \geq \alpha \ast \gamma / \alpha \), then \( I(\beta, \alpha) \ast \gamma \leq I(\beta, \alpha \ast \gamma) \).
• Case 2: \( \gamma \leq \alpha \leq \beta, \beta \neq 0 \). Since \( I(\beta, \alpha) \ast \gamma = \alpha \ast \gamma / \beta \) and \( I(\beta, \alpha \ast \gamma) = \alpha \ast \gamma / \beta \), then \( I(\beta, \alpha) \ast \gamma \leq I(\beta, \alpha \ast \gamma) \).
• Case 3: \( \beta \leq \gamma \leq \alpha, \beta \neq 0 \). Since \( I(\beta, \alpha) \ast \gamma = \gamma \) and \( I(\beta, \alpha \ast \gamma) \geq \min(\alpha \ast \gamma / \beta, 1) \geq \alpha \ast \gamma / \alpha \), then \( I(\beta, \alpha) \ast \gamma \leq I(\beta, \alpha \ast \gamma) \).
• Case 4: \( \beta \leq \alpha \leq \gamma, \beta \neq 0 \). Since \( I(\beta, \alpha) \ast \gamma = \alpha \ast \gamma / \beta \) and \( I(\beta, \alpha \ast \gamma) = \alpha \ast \gamma / \beta \), then \( I(\beta, \alpha) \ast \gamma \leq I(\beta, \alpha \ast \gamma) \).
• Case 5: \( \alpha \leq \gamma \leq \beta, \beta \neq 0 \). Since \( I(\beta, \alpha) \ast \gamma = \alpha \ast \gamma / \beta \) and \( I(\beta, \alpha \ast \gamma) = \alpha \ast \gamma / \beta \), then \( I(\beta, \alpha) \ast \gamma \leq I(\beta, \alpha \ast \gamma) \).
• Case 6: \( \alpha \leq \beta \leq \gamma, \beta \neq 0 \). Since \( I(\beta, \alpha) \ast \gamma = \alpha \ast \gamma / \beta \) and \( I(\beta, \alpha \ast \gamma) = \alpha \ast \gamma / \beta \), then \( I(\beta, \alpha) \ast \gamma \leq I(\beta, \alpha \ast \gamma) \).

This result has been generalized in [31] from our initial work at [32]. According to the theoretical results of [31], \( \otimes(I(\beta, \alpha), \gamma) \leq I(\beta, \otimes(\alpha, \gamma)) \) is always verified when dealing with residuated implications. In fact, Mamdani–Larsen operators also fulfills this law, as well as some cases of strong implications and Quantum and Dishkant operators [33].

3.4. Soundness of the proposed operators in relation with the semantics of division queries

We have introduced two inclusion operators that are able to compute approximate degrees of inclusion in those cases where the elements of the universe \( U \) are imprecise objects among which classical equality cannot be applied and, instead, a resemblance relationship must be used. Furthermore, these inclusion operators have been presented in order to interact with fuzzy object databases by means of queries that are similar to relational division queries.

In the fuzzy relational database area, the topic of flexible division induces that the devised operator has the basic property of a division, i.e., its result may be characterized as a maximal relation whose product with the divisor is included in the dividend.

Therefore, in order to guarantee the suitability of the proposed operators for the kind of queries we have described in Section 2.3, we must check that the use of operators \( \Theta \) and \( \Psi \) permits us to fulfill this relational law conveniently adapted to the soft universe \( U \) we deal with in fuzzy object databases.
Consider that we are using $\Theta$ as inclusion operator to solve the mentioned query (see again Figs. 3 and 4). According to Eq. (3), in an object oriented context, the dividend $t_d$ can be organized as a collection of the form $\{(a_1, A_1), \ldots , (a_p, A_p)\}$ where

- $a_1 \ldots a_p$ are the elements of the candidate set.
- $A_i = \mu_{A_i}(\text{object}_{i,1})/\text{object}_{i,1} + \cdots + \mu_{A_i}(\text{object}_{i,n_i})/\text{object}_{i,n_i}$ is the fuzzy set of objects to which $a_i$ is related in $t_d$.

Similarly, the divisor $t_b$ has the following form:

$$ t_b = \mu_{t_b}(\text{object}_{b,1})/\text{object}_{b,1} + \cdots + \mu_{t_b}(\text{object}_{b,k})/\text{object}_{b,k} $$

Finally, the result of the division-like query is a set of the form:

$$ t_d = \Theta_S(A_1|t_b)/a_1 + \cdots + \Theta_S(A_p|t_b)/a_p $$

The cartesian product of $t_b$ and $t_d$ is

$$ t_d \times t_b = \otimes(\Theta_S(A_1|t_b), \mu_{t_b}(\text{object}_{b,1}))/\text{object}_{b,1} + \cdots + \otimes(\Theta_S(A_p|t_b), \mu_{t_b}(\text{object}_{b,k}))/\text{object}_{b,k} $$

where $\otimes$ is the t-norm used to generate $\Theta$.

We can proceed with this cartesian product in a similar way to the characterization of the dividend in Section 2.3. Thus, for every candidate element $a_i$, we can define a fuzzy set of objects $R_i$ with the following membership function:

$$ \mu_{R_i}(\text{object}_j) = \mu_{t_d \times t_b}(a_i, \text{object}_j) $$

That is, each $R_i$ has the following form:

$$ R_i = \otimes(\Theta_S(A_i|t_b), \mu_{t_b}(\text{object}_{b,1}))/\text{object}_{b,1} + \cdots + \otimes(\Theta_S(A_i|t_b), \mu_{t_b}(\text{object}_{b,k}))/\text{object}_{b,k} $$

Once we have described the cartesian product result for every $a_i$ in the candidate set, we have to demonstrate that

$$ \forall i, R_i \subseteq A_i \tag{14} $$

At this point, we cannot forget that we are working on a referential set of objects $\mathcal{U}$ where equality is substituted by resemblance, and conventional inclusion has no sense, giving way to a resemblance driven inclusion as the represented by $\Theta$.

Thus, what we have to impose is that $R_i$ is included in $A_i$ at least at the same degree than $t_b$ is included in $A_i$, according to our resemblance driven inclusion operator.

That is, restriction of Eq. (14) should be transformed as follows:

$$ \forall i, \Theta_S(A_i|t_b) \leq \Theta_S(A_i|R_i) \tag{15} $$

In order to prove this restriction for every $a_i$, we have to consider that:

- $R_i \subseteq t_b$ That is $\forall j, \mu_{R_i}(\text{object}_{b,j}) \leq \mu_{t_b}(\text{object}_{b,j})$, which is straight forward due to $\otimes(\Theta_S(A_i|t_b), \mu_{t_b}(\text{object}_{b,j})) \leq \mu_{t_b}(\text{object}_{b,j})$, $\forall j$.
- $X_1 \subseteq X_2 \Rightarrow \Theta_S(X_3|X_1) \geq \Theta_S(X_3|X_2)$ Property 2 of $\Theta$ operator (see Section 3.1).

With respect to the maximality condition, we have to analyze the behavior of the proposed operator with respect to increments in the membership degrees of the result $t_d$. Consider a $t'_d$ of the form

$$ \forall i, \mu_{t'_d}(a_i) \geq \mu_{t_d}(a_i) \tag{16} $$

For each candidate element $a_i$, we can build a $R'_i$ from $t'_d \times t_b$ similarly as we proceeded with $R_i$ from $t_d \times t_b$

$$ \mu_{R'_i}(\text{object}_j) = \mu_{t'_d \times t_b}(a_i, \text{object}_j) \tag{17} $$

Once we have described the modified cartesian product result for every $a_i$ in the candidate set, we have to demonstrate that

$$ \forall i, R'_i \not\subseteq A_i \tag{18} $$
As we are working on a referential set of objects \( U \) where equality is substituted by resemblance, this conventional restriction has no sense. However, we can consider as alternative constraint that this new \( R'_i \) is not included in the corresponding \( A_i \) in a higher degree than the obtained \( R_i \). That is

\[
\forall i, \Theta_S(A_i|R'_i) \leq \Theta_S(A_i|R_i)
\]

(19)

In order to prove this restriction for all \( a_i \), we have to consider that:

- \( R_i \subseteq R'_i \). That is \( \forall j, \mu_{R_i}(\text{object}_{b,j}) \leq \mu_{R'_i}(\text{object}_{b,j}) \), which is straight forward due to Eq. (16) and the properties of t-norms.
- \( X_1 \subseteq X_2 \Rightarrow \Theta_S(X_3|X_1) \geq \Theta_S(X_3|X_2) \) Property 2 of \( \Theta \) operator (see Section 3.1).

Property 5 permits a similar argumentation in relation to \( \Psi \) operator.

Finally, in order to complete our analysis of the soundness of the proposed operators in relation with the semantics of division queries, it is important to point out that, in the case we use equality as resemblance operator, properties 1 and 4 state that both \( \Theta \) and \( \Psi \) coincide with the fuzzy division operator described in Eq. (2), which is proven to fulfil the desired conventional law [18].

4. The role of cardinality

In order to finish our analysis of inclusion operators between fuzzy sets of fuzzily described objects, this section pays attention to cardinality. Cardinality may play a relevant role when computing the inclusion for two main reasons:

- On the one hand, if we soften the inclusion so that similar objects can fulfil the implication constraint, we have to take into account the number of involved objects; that is, the same object \( y \) of \( A \) can be matched to more than one object \( x \) of \( B \) when computing:
  - \( \max_{y \in \mathbb{W}} \theta_{B,A,S}(x, y) \) in \( \Theta \),
  - and \( \max_{y \in \mathbb{W}} \psi_{B,A,S}(x, y) \) in \( \Psi \).

Section 4.1 is devoted to analyze this problem in depth.

- On the other hand, another way of softening the results is to be tolerant with the number of objects of set \( B \) included in \( A \). Section 4.2 studies this point.

4.1. Cardinality during the computation of \( \Theta \) and \( \Psi \)

As we have commented in the previous paragraphs, the use of resemblance relations instead of equality when computing inclusion degrees may make the obtained value be high, even if there is a great difference in terms of cardinality between the set of elements of \( B \) and the subset of elements of \( A \) matched to them during the computation of inclusion.

In some situations, cardinality may not be important. For instance, in the example of the construction database, in order to compute the solution of the query To find those companies whose machinery can carry out the set of tasks that can be done with all the listed machinery of a given project, for each construction machinery demanded in the project, we look for a construction machinery in the company with similar capabilities. In this context, it does not matter if the same construction machinery of the company is matched to more than one construction machinery demanded in the project. That is, the number of selected construction machineries of the company is not relevant, but the set of tasks they can carry out. It is possible that a given machine can substitute some others.

However, if it is important that the demanded machineries in the project can work in parallel, then, the number of selected construction machineries of the company is relevant and cardinality is important.

If we want to take cardinality into account, we can enclose to the inclusion degree a factor that takes into account the distance between the cardinalities of the sets that are being compared.


Table 2
Resemblance relation.

<table>
<thead>
<tr>
<th></th>
<th>object₁</th>
<th>object₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>object₁</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>object₂</td>
<td>1.0</td>
<td></td>
</tr>
</tbody>
</table>

**Definition 3 (Cardinality Factor).** \( CF : N \times N \rightarrow [0, 1] \) is a cardinality factor if:

1. If \( X = Y \), \( CF(X, Y) = 1 \)
2. If \( X \geq Y \), \( CF(X, C) \geq CF(Y, C) \)
3. If \( X \geq Y \), \( CF(C, Y) \geq CF(C, X) \)

A simple example of CF is the following one:

\[
CF(X, Y) = \begin{cases} 
1 & \text{if } X \geq Y \\
X/Y & \text{otherwise} 
\end{cases} 
\] (20)

We can consider the following definitions so that the Cardinality Factor can be taken into account together with \( \Theta \) and \( \Psi \).

**Definition 4 (Inclusion matching induced by \( \Theta_{B,A,S} \)).** Let \( A \) and \( B \) be two fuzzy sets defined over a finite reference universe \( \mathcal{U} \), \( S \) be a resemblance relation defined over the elements of \( \mathcal{U} \), \( \otimes \) be a t-norm, and \( I \) an implication defined according to \( \otimes \). Then \( IM_{\Theta_{B,A,S}} \subseteq \mathcal{U} \times \mathcal{U} \) is an inclusion matching induced by \( \Theta_{B,A,S} \) if:

- \( \forall (x, y) \in IM_{\Theta_{B,A,S}}, x \in support(B) \)
- \( \forall (x, y) \in IM_{\Theta_{B,A,S}}, y' \in \mathcal{U}, \Theta_{B,A,S}(x, y') > \Theta_{B,A,S}(x, y) \)
- \( \forall x \in support(B), \exists (x, y) \in IM_{\Theta_{B,A,S}} \)
- \( \exists (x, y) \) and \( (x, y') \in IM_{\Theta_{B,A,S}}, y \neq y' \)

\( \{< object₁, object₁ >, < object₂, object₁ > \} \) is an inclusion matching induced by \( \Theta_{B,A,S} \) in the example of Fig. 9 and Table 2. This means that, during the computation of \( \Theta \), for \( object₁ \in support(B) \), \( object₁ \in support(A) \) maximizes the value of \( \Theta_{B,A,S} \), and for \( object₂ \in support(B) \), \( object₁ \in support(A) \) maximizes the value of \( \Theta_{B,A,S} \). Notice that this inclusion matching entails that the same object of \( A \) is being used to satisfy the inclusion of the two objects of \( B \).

Similarly, we can define the inclusion matching induced by \( \Psi_{B,A,S} \):

**Definition 5 (Inclusion matching induced by \( \Psi_{B,A,S} \)).** Let \( A \) and \( B \) be two fuzzy sets defined over a finite reference universe \( \mathcal{U} \), \( S \) be a resemblance relation defined over the elements of \( \mathcal{U} \), \( \otimes \) be a t-norm, and \( I \) an implication defined
according to $\otimes$. Then $IM_{\psi_{B,A,S}} \subseteq \mathbb{U} \times \mathbb{U}$ is an inclusion matching induced by $\psi_{B,A,S}$ if:

- $\forall (x, y) \in IM_{\psi_{B,A,S}}, x \in \text{support}(B)$
- $\forall (x, y) \in IM_{\psi_{B,A,S}}, \exists y' \in \mathbb{U}, \psi_{B,A,S}(x, y') > \psi_{B,A,S}(x, y)$
- $\forall x \in \text{support}(B), \exists (x, y) \in IM_{\psi_{B,A,S}}$
- $(x, y)$ and $(x', y') \in IM_{\psi_{B,A,S}}$, $y \neq y'$

$\{< \text{object}_1, \text{object}_1 >, < \text{object}_2, \text{object}_1 >\}$ is also an inclusion matching induced by $\psi_{B,A,S}$ in the example of Fig. 9 and Table 2.

When querying the system using $\Theta$ (similarly with $\Psi$), once we have chosen the CF function we want to use, the user can require to enclose the following factor to the obtained inclusion degree:

$$CF(MC\theta_S(A|B), |\text{support}(B)|)$$

where

$$MC\theta_S(A|B) = \sup\{|IM|, \text{IM is and inclusion matching induced by } \theta_{B,A,S}\}.$$ 

It is important to remark that there may exist more than one inclusion matching induced by $\theta_{B,A,S}$, since it is possible that two objects, namely, $\text{object}_j$ and $\text{object}_k \in \text{support}(A)$, maximize the value of $\theta_{B,A,S}$ (respectively, $\psi_{B,A,S}$) for a given $\text{object}_l \in \text{support}(B)$. That is the reason to use $MC\theta_S(A|B)$ in the above formulation.

Using (20) in the example of Fig. 9, the resemblance relation of Table 2, the minimum as t-norm and Gödel as implication, we have:

$$\Theta_S(A|B) = 1.0$$

$$\Psi_S(A|B) = 1.0$$

but $CF(MC\Theta_S(A|B), |\text{support}(B)|) = 0.5$ and $CF(MC\Psi_S(A|B), |\text{support}(B)|) = 0.5$.

This way of taking cardinality into account is simple and easy to compute. Nevertheless, more sophisticated ways of considering cardinality can be used. For example, operators $\Theta$ (Definition 1) and $\Psi$ (Definition 2) can be modified so that they do not use the same element of $A$ to match more than one element of $B$. The computation of these new versions of the operators are more expensive from the computational point of view and are out of the scope of this paper. For the interested reader, some operators with this semantics for the case of simple objects can be found in [34].

4.2. Cardinality as a relaxation of the inclusion

The other role cardinality can play in an inclusion operator is to soften the amount of elements of $B$ that have to fulfil the inclusion relation. That is, with this kind of relaxation, the idea is to compute to what extent almost all elements of $B$ are included in $A$. For example, in the case of the construction database, the query could be To find those companies which can work with almost all the listed machinery of a given project.

This approach has been widely used in the context of fuzzy relational database models giving rise to interesting results. The basic idea is to allow that not all the elements of a fuzzy set hold a concrete property, but by a part or, more exactly, by a proportion of elements. This proportion is, at the same time, formulated in a flexible way by means of a linguistic quantifier as “almost all”, “most of them”, “many”,... and the fulfillment degree of the considered quantified sentence must be calculated.

There exist in the literature different ways to compute such fulfillment degree. The first paper about this problem (authored by Yager [16]) makes use of the OWA operator to aggregate the fulfillment degrees of the elements according to a considered quantifier. In [17,35] the authors give a simpler approach based on the idea of aggregating with only two values, which summarize the quantifier.

More recently, some authors have proposed other approaches since this is a widely studied problem. Among these papers, we would like to mention [20,18,21] where an interesting analysis of the different approaches to relational division is made.
Anyway, as we have mentioned before, the problem of softening the inclusion condition can be formulated as the problem of computing the fulfillment degree of the sentence: *Q elements of B are included in A*, where to be included is an imprecise property measured by means of the resemblance operators defined in Section 3.

Once established this property, the aggregation of the obtained values according to Q quantifier, can be carried out by means of any of the approaches for the computation of quantified sentences in the literature, either those ones based on some form of cardinality of a fuzzy set (e.g. [36,37]) or those ones oriented to weights (e.g. [35,38] with the OWA definition or [39] with OWA generalizations).

In order to formalize these ideas, we give the following definition:

**Definition 6 (Inclusion softened by quantifiers).** Let A and B be two fuzzy sets of objects defined over a finite reference set ℱ, S be a resemblance relation defined over the elements of ℱ, ⊗ a t-norm, and I an implication function defined according to it. Let us also consider a linguistic quantifier Q and an aggregation procedure \( Agg_Q(C(x)) \) such that when applied to a fuzzy set C of reference set X, it permits us to obtain the fulfillment degree of the sentence “Q of X are C”.

In this context, we define the following concepts:

- Quantified \( Θ \): \( Θ_{Q,S}(A|B) = Agg_Q(\max_{B,A,S}(x), x \in support(B)) \)
- Quantified \( Ψ \): \( Ψ_{Q,S}(A|B) = Agg_Q(\max_{B,A,S}(x), x \in support(B)) \)

where

- \( ∀x \in ℱ, \max_{B,A,S}(x) = \max_{y \in ℱ}(\min_{B,A,S}(x, y)) \)
- \( ∀x \in ℱ, \max_{B,A,S}(x) = \max_{y \in ℱ}(\min_{B,A,S}(x, y)) \)

Considering the properties of procedures for computing quantified sentences, we can assure that: \( Θ_{all,S}(A|B) = Θ_S(A|B) \) and \( Ψ_{all,S}(A|B) = Ψ_S(A|B) \).

As aggregation procedures of the form \( Agg_Q(C(x)) \) always fulfills this law

\[
∀C, C'|C' \subseteq C, Agg_Q(C'(x)) ≤ Agg_Q(C(x))
\]

then, properties 2 and 5 also hold for \( Θ_{Q,S}(A|B) \) and \( Ψ_{Q,S}(A|B) \), respectively. Thus, the considerations of Section 3.4 are also valid to prove that \( Θ_{Q,S}(A|B) \) and \( Ψ_{Q,S}(A|B) \) are compliant with the semantic of the division.

### 5. Implementation issues in pg4DB

As we have commented in Section 1, one of the main goals of our research is that the proposed theoretical framework can be easily used through sql99-compliant sentences in a fuzzy object database. Therefore, in this section, we show the extended query capabilities that have been implemented in pg4db [12] in order to be able to use the operators proposed in previous sections. To do that, we take the example of construction database introduced at the end of Section 2.3. Additionally, we end the section analyzing some important notes concerning the implementation of the proposed capabilities.

#### 5.1. Division-like queries in pg4BD

The system pg4DB (POSTgreSQL Fuzzy Object Relational DataBase) is a framework that supports the management of fuzzy objects in object-relational database management systems such as PostgreSQL [12]. pg4DB allows users to create imprecise objects and operate with them using SQL99-compliant sentences. One of the main advantages of pg4DB is that it allows the user to create fuzzy object tables with a configurable built-in comparison logic able to manage the fuzziness.

System pg4DB has been extended so that operators \( Θ \) and \( Ψ \) can be used. In this section, we present some queries for the example of construction database as a proof of concept of the above mentioned operators. The example database has, among other, three object tables:

- **MACHINE**: objects of this table describe the different machineries that are registered by the promoter. Machinery objects are complex and fuzzily described. Several comparison profiles are defined for this class.
Fig. 10. Offered and Demanded Machinery in the example database. (a) Offered_machinery attribute in Company. (b) Demanded_machinery attribute in Project.

- COMPANY: objects of this table describe the different companies of the database. One of the attributes, offered_machinery, is a fuzzy set of objects of class MACHINERY. Within the domain of this attribute, by default, MACHINERY objects are fuzzily compared from a construction perspective.

- PROJECT: objects of this table describe the information of the projects that are registered by the promoter. Attribute demanded_machinery is a fuzzy set of objects of class MACHINERY that contains the list of machineries needed in the project together with the demanded expertise level. As is the case of the attribute offered_machinery, within the domain of this attribute, by default, MACHINERY objects are compared from a construction perspective.

We can query the system so that a list of the registered machinery is shown. The extended SQL statement is as follows:

```sql
SELECT m1.name as MACHINERY1, m2.name as MACHINERY2, 
     FEQ(m1.oid, m2.oid, 'CONSTRUCTION') as FEQ
FROM machinery as m1, machinery as m2
WHERE m1.name < m2.name;
```

<table>
<thead>
<tr>
<th>MACHINERY1</th>
<th>MACHINERY2</th>
<th>FEQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cement Mixer Truck</td>
<td>Crane truck</td>
<td>0.2</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cement Mixer Truck</td>
<td>Static Cement Mixer</td>
<td>0.5</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crane truck</td>
<td>Mobile Crane</td>
<td>0.7</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crane Truck</td>
<td>Static Cement Mixer</td>
<td>0.0</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This query launches the recursive comparison process between each couple of machineries. The label CONSTRUCTION indicates the comparison profile (previously defined by the user) that has to be used (basically, a comparison profile determines the importance weights of every attribute and the operators to be used in the comparison process described in Section 2).

These resemblance values between machinery objects can be used in division queries like the ones expressed in our example. As operators have been deeply analyzed in the previous sections, next queries are mainly aimed to show the sql99 sentences used in pg4DB in order to deal with them. In any case, Fig. 10(a) and (b) shows relevant data useful to follow the computations (the whole dataset is omitted here for the sake of space).

\(^1\) FEQ is an operator that computes the resemblance between objects fuzzily described following the approach described in Section 2.2. For the particular implementation of the operator see [12].
For instance, in order to find the companies that accomplish all the machinery demand of a given project ‘P1’, we only have to write\(^2\):

```
SELECT c.name AS NAME,
   THETA(c.offered_machinery,p.demanded_machinery) AS THETA
FROM company AS c, project AS p
WHERE p.code='P1';
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>THETA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfacar Buildings</td>
<td>0.7</td>
</tr>
<tr>
<td>Hita Marin</td>
<td>1.0</td>
</tr>
<tr>
<td>Lopez and Sons</td>
<td>0.5</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

By default, no cardinality factor is used. However, we can query the cardinality factor value described in Section 4:

```
SELECT c.name AS NAME,
   CF_THETA(c.offered_machinery,p.demanded_machinery) AS CF_THETA
FROM company AS c, project AS p
WHERE p.code='P1';
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>CF_TH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfacar Buildings</td>
<td>0.5</td>
</tr>
<tr>
<td>Hita Marin</td>
<td>1.0</td>
</tr>
<tr>
<td>Lopez and Sons</td>
<td>1.0</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Similar queries can be written using the \(\Psi\) semantics:

```
SELECT c.name AS NAME,
   PSI(c.offered_machinery,p.demanded_machinery) AS PSI
FROM company AS c, project AS p
WHERE p.code='P1';
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>PSI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfacar Buildings</td>
<td>1.0</td>
</tr>
<tr>
<td>Hita Marin</td>
<td>1.0</td>
</tr>
<tr>
<td>Lopez and Sons</td>
<td>0.5</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

```
SELECT c.name AS NAME,
   CF_PSI(c.offered_machinery,p.demanded_machinery) AS CF_PSI
FROM company AS c, project AS p
WHERE p.code='P1';
```

<table>
<thead>
<tr>
<th>NAME</th>
<th>CF_PS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alfacar Buildings</td>
<td>0.5</td>
</tr>
<tr>
<td>Hita Marin</td>
<td>1.0</td>
</tr>
<tr>
<td>Lopez and Sons</td>
<td>1.0</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Finally, we can query the system with a Vila aggregation operator \([17,35]\), for example based on the quantifier \(\text{MOST}\), instead of the predefined \(\Psi\), according to the proposal of Definition 6:

```
SELECT c.name AS NAME,
   VQ_PSI(c.offered_machinery,p.demanded_machinery,‘MOST’) AS VQ_PSI
FROM company AS c, project AS p
WHERE p.code='P1';
```

\(^2\) Gödel and min are used by default.
The same query, with the quantifier \textit{ALL}, delivers the same result that the one that uses the built-in \textit{Ψ} operator:

\begin{verbatim}
SELECT c.name AS NAME,
    VQ_PSI(c.offered_machinery,p.demanded_machinery,'ALL') AS VQ_PS
FROM company AS c, project AS p
WHERE p.code='P1';
\end{verbatim}

5.2. Some notes about computational complexity

Though the description of the implementation of these capabilities in pg4db is out of the scope of this paper, there are some important ideas related to this implementation that deserve special attention:

- The comparison process described in Section 2.2 is recursive: the computation of the resemblance of two complex objects \(o_1\) and \(o_2\) involves the computation of the comparison of their attributes values that, in turn, can be objects and require the initialization of a new recursive comparison process. In this process, we have to deal with the potential presence of cycles in the data graph (i.e. we might have to compute the resemblance of objects \(o_1\) and \(o_2\) in order to compute the resemblance between objects \(o_1\) and \(o_2\)) \cite{28}. Additionally, in order to speed up this computation, a cache is generated where computed values of FEQ that arise during the comparison process are stored. The use of this cache avoids the re-computation of FEQ values when needed.

- When computing \(\Theta\) (similarly with \(\Psi\)), for each object \(x \in U\) we have to scour the whole \(U\) in order to find the object \(y\) that maximizes the value of \(\Theta\). Actually, only elements of \(support(B)\) are considered in the role of \(x\) and only elements of \(support(A)\) are considered in the role of \(y\); to explore other combinations is useless. Once again, the use of the above mentioned cache considerably reduces the computation effort.

- If we do not allow that the same object \(y\) is matched to more than one object \(x\) during the computation of the inclusion, we have to deal with different sets of operators (like extensions of the one proposed in \cite{34}). This operators have to explore the whole set of possible matchings with the increase of complexity that this fact involves. Operators \(\Theta\) and \(\Psi\) are conceived without imposing this restriction. In case this cardinality issue is important for the user, we propose the use of a cardinality factor (as discussed in Section 4.1) that, though involving additional computations, its cost is not as high as the cost to explore of all the possible permutations.

6. Conclusions

In this paper, the resolution of queries that are similar to the division queries of the relational data model is analyzed from an object-oriented perspective. The fact that attribute values can be fuzzily described objects makes us to extend the division operator by means of resemblance measures. We propose an approach that considers two ways of incorporating such resemblance measures in the computation of inclusion, namely: as a constraint of the implication and as a constraint of the membership to the dividend. We have studied two operators that implement these two inclusion measures.

Additionally, we have analyzed the role that cardinality can play in this kind of queries. In this sense, we have considered to attach a cardinality factor to the inclusion operators that takes into account the number of elements of set \(A\) matched during the inclusion analysis.

Finally, we have also proposed two approximate inclusion operators based on the use of a quantifier for relaxing the division condition.
With the described survey of operators, the user can semantically adapt the division operators to her/his needs in an object-oriented context. A proof of concept implementation of the operators in pg4DB has been presented as a demonstration of the proposed query capabilities.

Acknowledgments

Work partially supported by the projects Representación y Manipulación de Objetos Imperfectos en Problemas de Integración de Datos: Una Aplicación a los Almacenes de Objetos de Aprendizaje, Junta de Andalucía, Consejería de Economía, Innovación y Ciencia (P07-TIC-03175), and Tratamiento de Información Difusa Temporal en un SGBD Relacional: Formulación Teórica, Implementación y Aplicaciones, Estado Español (TIN2008-02066).

References


