Querying Fuzzy Relational Databases Through Fuzzy Domain Calculus

José Galindo,[1,*] Juan Miguel Medina,[2] Mª Carmen Aranda[3]
[1]Departamento Lenguajes y Ciencias de la Computación, Universidad de Málaga, E.T.S.I. Informática 29071 Málaga, Spain
[2]Departamento Ciencias de la Computación e I.A., Universidad de Granada, Av Andalucia no 38, 18071 Granada, Spain

In this paper we present a definition of a domain relational calculus for fuzzy relational databases using the GEFRED model as a starting point. It is possible to define an equivalent fuzzy tuple relational calculus and consequently we achieve the two query language levels that Codd designed for relational databases but these are extended to fuzzy relational databases: Fuzzy relational algebra (defined in the GEFRED model) and the fuzzy relational calculus which is put forward in this paper. The expressive power of this fuzzy relational calculus is demonstrated through the use of a method to translate any algebraic expression into an equivalent expression in fuzzy domain relational calculus. Furthermore, we include a useful system so that the degree to which each value has satisfied the query condition can be measured. Some examples are also included in order to clarify the definition. © 1999 John Wiley & Sons, Inc.

I. INTRODUCTION

The relational database model was developed by E. F. Codd of IBM and published in 1970 in Ref. 1. In addition, Codd designed two levels of data manipulation languages (DML) in Ref. 2: relational algebra and relational calculus.

Relational calculus uses first-order predicate calculus. Although the idea of using the predicate calculus as a basis for a query language seems to have originated in an article by Kuhns,[3] it was originally proposed for application to relational databases, as we said above, in Ref. 2. In this article, Codd also introduces the concept of relational completeness: a query language is relationally complete if every query that may be written by means of calculus expressions, may also be written using propositions in that language. In the article, Codd also describes "Codd's reduction algorithm" to translate any calculus

*Author to whom correspondence should be addressed. e-mail: ppgg@lcc.uma.es.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 14, 375–411 (1999)

CCC 0884-8173/99/040375-37

expression into a semantically equivalent algebraic expression, demonstrating the relational algebra completeness and giving a possible way to compute the calculus.

Codd's calculus is known as tuple calculus because it is based on the tuple variable concept, which is a variable that takes values from tuples of a relation or from the union of two or more relations, i.e., the only permitted values for a tuple variable are the tuples of a single relation or of a union.

One implementation of tuple calculus was the QUEL language, used in the INGRES system.[4] The SQL language[5,6] has some calculus elements too.

In 1977, Lacroix and Pirotte proposed in Ref. 7 an alternative relational calculus, the domain calculus, where the tuple variables are changed by domain variables, which take values on the underlying domain of the attributes of the relations. The same authors presented in Ref. 8 a language based on this calculus, ILL. Pirotte and Wodon presented in Ref. 9 the FQL language, which is far more formal than ILL. Query by example (QBE), presented by Zloof in Ref. 10 and other articles, may also be considered as a particular form of the domain calculus.

Definitions of relational algebra and relational tuple and domain calculus may be seen many times in the bibliography.[11–14] In Ref. 14 Ullman shows how to translate an algebraic expression into a tuple calculus expression, how to translate a tuple calculus expression into a domain calculus expression, and how to translate a domain calculus expression into an algebraic expression.

These query languages are perfectly adapted to classic relational databases with all the domains being crisp. In fuzzy databases, it is more difficult for the definitions to cover a lot of cases. In Refs. 15 and 16, the GEFRED model was proposed for fuzzy relational databases, giving a fuzzy relational algebra for this model. The GEFRED model represents a synthesis among the different models which have appeared to deal with the problem of the representation and management of fuzzy information in relational databases. One of the main advantages of this model is that it consists of a general abstraction which allows us to deal with different approaches, even when these may seem disparate.

Here, we present a definition of fuzzy domain relational calculus framed in the GEFRED model, although its theoretic basis may be used in other models.

In Ref. 17 a domain calculus is proposed for Buckles–Petry's fuzzy relational database model,[18,19] which is much more restrictive than the GEFRED model.

We choose domain calculus instead of tuple calculus because domain calculus is more explicit since it uses domain variables and manages each attribute independently. Therefore, domain calculus tends to be closer to natural language than tuple calculus.

The fundamental difference between tuple calculus and domain calculus lies basically in how the user perceives the database, i.e., the relations and the attributes. In tuple calculus the main entities are the relations which have various properties (its attributes). In domain calculus the main entities are the attributes, which have relations among them represented by database relations.

This latter vision is closer to how humans see and understand the universe represented by the database.

First, we give some preliminary concepts about GEFRED; then we give the definition of the fuzzy domain relational calculus with the calculus expressions for algebraic primitive operators and for the more useful nonprimitive operators and prove that any algebraic expression may be translated into an equivalent calculus expression. We will then show a mechanism to ascertain, in the resulting relation, the fulfilment degree to which each value has satisfied the query condition. Finally, we explain some practical examples to show the expressive power of this language and how to extract the fulfilment degree in the resulting relation.

II. PRELIMINARY CONCEPTS ABOUT GEFRED

The GEFRED model is based on the definition which is called *generalized fuzzy domain* (D) and *generalized fuzzy relation* (R), which include classic domains and classic relations, respectively.

DEFINITION 2.1. *If U is the discourse or universe, $\tilde{\mathscr{P}}(U)$ is the set of all possibility distributions defined for U, including those which define the* Unknown *and* Undefined *types* (*types* 8 *and* 9 *in Table* I), *and* NULL *is another type defined in Table* I (*type* 10) *therefore, we define the generalized fuzzy domain as $D \subseteq \tilde{\mathscr{P}}(U) \cup \mathrm{NULL}$.*

The Unknown, Undefined, *and* NULL *types are defined according to Umano,*[20] *and Fukami et al.*[21]

With these fuzzy domains, all data types can be represented in Table I.

Table I. Data types.

1.	A single scalar (e.g., Size = Big, represented by the possibility of distribution 1/Big).
2.	A single number (e.g., Age = 28, represented by the possibility of distribution 1/28).
3.	A set of mutually exclusive possible scalar assignations (e.g., Behavior = {Bad, Good}, represented by {1/Bad, 1/Good}).
4.	A set of mutually exclusive possible numeric assignations (e.g., Age = {20, 21}, represented by {1/20, 1/21}).
5.	A possibility distribution in a scalar domain (e.g., Behavior = {0.6/Bad, 1.0/Regular}).
6.	A possibility distribution in a numeric domain (e.g., Age = {0.4/23, 1.0/24, 0.8/25}, fuzzy numbers or linguistic labels).
7.	A real number belonging to [0, 1], referring to the degree of matching (e.g., Quality = 0.9).
8.	An Unknown value with possibility distribution. Unknown = $\{1/d\colon e \in D\}$ on domain D, considered.
9.	An Undefined value with possibility distribution. Undefined = $\{0/d\colon d \in D\}$ on domain D, considered.
10.	A NULL value given by NULL = {1/Unknown, 1/Undefined}.

DEFINITION 2.2. *A generalized fuzzy relation, R, is given by two sets: "Head" ($\mathscr{H}$) and "Body" ($\mathscr{B}$), $R = (\mathscr{H}, \mathscr{B})$, defined as:*

- *The Head consists of a fixed set of attribute-domain-compatibility terms (where the last is optional),*

$$\mathscr{H} = \{(A_1 \colon D_1[, C_1]), (A_2 \colon D_2[, C_2]), \dots, (A_n \colon D_n[, C_n])\}$$

where each attribute A_j has an underlined fuzzy domain, not necessarily different, D_j ($j = 1, 2, \dots, n$). C_j is a "compatibility attribute" which takes values in the range $[0, 1]$.

- *The Body consists of a set of different generalized fuzzy tuples, where each tuple is composed of a set of attribute-value-degree terms (the degree is optional),*

$$\mathscr{B} = \left\{\left(A_1 \colon \tilde{d}_{i1}[, c_{i1}]\right), \left(A_2 \colon \tilde{d}_{i2}[, c_{i2}]\right), \dots, \left(A_n \colon \tilde{d}_{in}[, c_{in}]\right)\right\}$$

with $i = 1, 2, \dots, m$, where m is the number of tuples in the relation, and where $\tilde{d}_{ij}$ represents the domain value for the tuple i and the attribute A_j, and c_{ij} is the compatibility degree associated with this value.

DEFINITION 2.3. *Let R be a generalized fuzzy relation expressed by*

$$R = \begin{cases} \mathscr{H} = \{(A_1 \colon D_1[, C_1]), \dots, (A_n \colon D_n[, C_n])\} \\ \mathscr{B} = \left\{\left(A_1 \colon \tilde{d}_{i1}[, c_{i1}]\right), \dots, \left(A_n \colon \tilde{d}_{in}[, c_{in}]\right)\right\} \end{cases} \tag{1}$$

with $i = 1, 2, \dots, m$, m being the number of tuples in the relation. Therefore, we may define:

- *Value component of a generalized fuzzy relation, R^v, as the part in the relation given by*

$$R^v = \begin{cases} \mathscr{H}^v = \{(A_1; D_1), \dots, (A_n \colon D_n)\} \\ \mathscr{B}^v = \left\{\left(A_1 \colon \tilde{c}_{i1}\right), \dots, \left(A_n \colon \tilde{d}_{in}\right)\right\} \end{cases} \tag{2}$$

where $\mathscr{H}^v$ and $\mathscr{B}^v$ are the value components of the "head" and "body," respectively.

- *Compatibility component of a generalized fuzzy relation, R^c, as the part in the relation given by*

$$R^c = \begin{cases} \mathscr{H}^c = \{[C_1], \dots, [, C_n]\} \\ \mathscr{B}^c = \{[c_{i1}], \dots, [, c_{in}]\} \end{cases} \tag{3}$$

where $\mathscr{H}^c$ and $\mathscr{B}^c$ are the compatibility components of the head and body, respectively.

The comparison operators are also redefined in order to adapt to the fuzzy nature of our data:

DEFINITION 2.4. *Let U be the considered discourse domain. We will call any fuzzy relation defined on U, the extended comparator θ, expressed in the following way,*

$$\begin{aligned} \theta\colon U \times U &\to [0,1] \\ \theta(u_i, u_j) &\mapsto [0,1] \end{aligned} \tag{4}$$

with $u_i, u_j \in U$.

DEFINITION 2.5. *Let U be the considered discourse domain, let D be the generalized fuzzy domain constructed on it, and let θ be an extended comparator defined on U. Let us consider a function Θ^θ defined as*

$$\begin{aligned} \Theta^\theta\colon D \times D &\to [0,1] \\ \Theta^\theta\left(\tilde{d}_1, \tilde{d}_2\right) &\in [0,1] \end{aligned} \tag{5}$$

Therefore, we may say that Θ^θ is a generalized fuzzy comparator on D induced by the extended comparator θ, if it verifies

$$\Theta^\theta\left(\tilde{d}_1, \tilde{d}_2\right) = \theta(d_1, d_2) \qquad \forall d_1, d_2 \in U \tag{6}$$

where $\tilde{d}_1, \tilde{d}_2$ represent the possibility distributions $1/d_1, 1/d_2$, induced by the values d_1, d_2, respectively.

On these definitions, GEFRED redefines the relational algebraic operators in the so-called generalized fuzzy relational algebra: union, intersection, difference, Cartesian product, projection, selection, join, and quotient. These operators are defined giving the head and body of a generalized fuzzy relation which is the result of the operation. All these operators are defined in Refs. 15 and 16, but the quotient is defined in Ref. 22.

In a relation (Definition 2.2), the compatibility degree for an attribute value (in a tuple) is obtained by manipulation processes performed on that relation and it indicates the degree to which that value has satisfied the operation performed on it.

III. FUZZY DOMAIN RELATIONAL CALCULUS

We use the classic domain relational calculus to define the fuzzy domain relational calculus, using the notation in Ref. 14. We first define the valid expressions in this relational calculus with the fuzzy atoms and the well formed formulas (WFF) with fuzzy atoms. Next, we look at how to restrict the calculus expressions to those that represent a finite relation ("safe" expressions).

A. Definition of Fuzzy Domain Calculus Expressions

The expressions, in fuzzy domain relational calculus, take the following form,

$$\{x_1, x_2, \ldots, x_n \mid \psi(x_1, x_2, \ldots, x_n)\}$$

where:

- $x_1, x_2, \ldots, x_n$ are domain variables, i.e., variables whose values are in the domain in which they are defined. These variables take values in the range (or domain) of a particular attribute in a generalized fuzzy relation. Consequently, these variables sometimes have the same name as the attribute. With these variables we express a tuple with the attributes we want in the resulting relation. The resulting relation will be a generalized fuzzy relation. Among the x_i there may also exist constants or expressions which use constants and variables, but these possibilities are not taken into account in order to simplify and to better focus the problem.
- $\psi(x_1, x_2, \ldots, x_n)$ is a well formed formula or WFF built with fuzzy atoms and with some specified operators. This formula must have $x_1, x_2, \ldots, x_n$ as the only free variables and it expresses a predicate or condition that must be satisfied by all the tuples in the resulting relation. A predicate ψ may only be True or False (remember that it is based on the first-order predicate calculus). Next, all these concepts are defined.

III.A.1. Fuzzy Atoms

We define the fuzzy atoms from formulas ψ (WFF) consisting of two parts:

(1) Fulfilment degree: Δ
(2) Fulfilment threshold: γ

and they will be expressed using the crisp comparator "$\geq$",

$$\Delta \geq \gamma$$

The fulfilment degree Δ of the atoms may also be of two types:

(1) Ownership: $\Delta = R(x_1, x_2, \ldots, x_n)$, where R is a generalized fuzzy relation of arity n and cardinality m [as in Eq. (2)] and each x_i is a constant or a domain variable. This atom expresses the fulfilment (or truth) degree of the affirmation which holds that $(x_1, x_2, \ldots, x_n)$ is a tuple of R. This fulfilment degree is computed as follows,

$$R(x_1, x_2, \ldots, x_n) = \max_{r=1,\ldots,m} \left\{ \min_{c=1,\ldots,n} \left\{ \Theta^{=}\left(\tilde{d}_{rc}, x_c\right) \right\} \right\} \tag{7}$$

where $\Theta^{=}$ is a generalized fuzzy comparator (see Definition 2.5) on D induced by the extended comparator $=(d, d') = \delta(d, d')$, and defined in general form as

$$\Theta^{=}(\tilde{p}, \tilde{p}') = \sup_{(p,p') \in U \times U} \min\left(=(p, p'), \pi_{\tilde{p}}(p), \pi_{\tilde{p}'}(p')\right) \tag{8}$$

$$= \sup_{d \in U} \min\left(\pi_{\tilde{p}}(d), \pi_{\tilde{p}'}(d)\right) \tag{9}$$

where $\tilde{p}, \tilde{p}' \in D$, and their associated possibility distributions are $\pi_{\tilde{p}}$ and $\pi_{\tilde{p}'}$, respectively. U is the discourse domain underlying the generalized fuzzy domain D (see Definition 2.1). The comparator $\Theta^{=}$ is translated into "approximately equal."

If we denote one tuple as $K = (k_1, \ldots, k_n)$ where all k_i are constants, then we can say that the jth tuple, $(\tilde{d}_{j1}, \ldots, \tilde{d}_{jn})$, is the most similar tuple to K in R so that it verifies

$$\max_{r=1,\ldots,m} \left\{ \min_{c=1,\ldots,n} \left\{ \Theta^{=}\left(\tilde{d}_{rc}, k_c\right) \right\} \right\} = \min_{c=1,\ldots,n} \left\{ \Theta^{=}\left(\tilde{d}_{jc}, k_c\right) \right\} \quad (10)$$

The most similar tuple to K in R is the R tuple $(\tilde{d}_{j1}, \ldots, \tilde{d}_{jn})$ which cost closely resembles $(k_1, \ldots, k_n)$. Therefore, $R(K)$ takes the value [in Eq. (7)] of the similarity degree between K and the tuple which is the most similar to it in R. In other words, Eq. (7) computes the greatest similarity between K and an R tuple (the tuple which is most similar to it). For means of simplification, this similarity which is calculated by Eq. (7) is sometimes called the ownership degree of a tuple in R.

(2) Comparison: $\Delta = \Theta^{\theta}(x, y)$, where Θ^{θ} is a generalized fuzzy comparator induced by the extended comparator θ, and x, y are constants or domain variables. This fuzzy atom expresses the fulfilment degree to which x is related to y by means of the comparator Θ^{θ}. These fuzzy atoms will be called fuzzy comparisons.

We can see that crisp comparisons are a particular case of fuzzy comparisons and they are included within them. Therefore, in order to clarify the calculus expressions we could use crisp comparisons with the traditional infixed notation: $x * y$, where x and y are constants or domain variables and $*$ is an arithmetic comparator: $* \in \{=, \neq, <, \leq, >, \geq\}$.

The fulfilment threshold γ is a real constant with $\gamma \in [0,1]$, which states a limit that Δ must overcome in order to consider the atom as True. We establish that it is possible to write atoms without the threshold γ when this is 1. This is very usual in crisp comparisons and in ownership atoms to indicate that we only consider those tuples in a concrete relation.

Example 1. Let the following be:

- R and S are generalized fuzzy relations of arity 4 and 3, respectively.
- x_1, x_2, and x_3 are domain variables.
- "Good" is a constant liguistic label with an associated possibility distribution.
- #n is the fuzzy number "approximately n," with n being a numeric constant.
- $\Theta^{>}$ is the generalized fuzzy comparator "greater than."
- $\Theta^{>>}$ is the generalized fuzzy comparator "very much greater than."

Then, in Table II we show some examples of fuzzy atoms and their meaning.

Note how in each fuzzy atom field it is possible to put domain variables, crisp constants, or fuzzy constants (fuzzy numbers, linguistic labels, possibility distributions ...). ■

Table II.

$R(x_1, x_2, 13, 8) \geq 0.6$	Tuple $(x_1, x_2, 13, 8)$ belongs to R with a minimum degree of 0.6
$S(x_1, Good, x_3)$	Tuple (x_1, Good, x_3) belongs to S with a minimum degree of 1
$\Theta^{=}(x_1, \text{Good}) \geq 0.9$	x_1 is Good with a minimum degree of 0.9
$\Theta^{>}(x_1, \#15) \geq 0.25$	x_1 is greater than "approximately 15" with a minimum degree of 0.25
$\Theta^{>>}(\#5, x_2) \geq 0.5$	"Approximately 5" is very much greater than x_2 with a minimum degree of 0.5

The fulfilment threshold meaning and utility vary depending on the atom type:

- Ownership atoms: In this atom type we apply a criterion in order to discover when two fuzzy tuples may be considered different. In Eq. (7) we give $R(x_1, \ldots, x_n)$ the similarity value of tuple $(x_1, \ldots, x_n)$ with the R tuple which is most similar to it. If this similarity is greater than or equal to the fulfilment threshold γ, i.e., if $R(x_1, \ldots, x_n) \geq \gamma$ then we consider that $(x_1, \ldots, x_n) \in R$. In this case the atom will be True and in any other case the atom will be False.
- Comparison atoms: The fulfilment threshold also states if the atom is considered true or false: if the fuzzy comparator value is greater than or equal to γ, then the atom will be True. In any other case the atom will be False.

III.A.2. Well Formed Formulas with Fuzzy Atoms

We now define the valid operators, free and bound occurrences of domain variables and the well formed formulas or WFF with fuzzy atoms. A WFF with fuzzy atoms is defined as those WFF of classic calculus but including fuzzy atoms:

(1) A fuzzy atom is a WFF. All its occurrences of domain variables are free.
(2) If ψ is a WFF, then $\neg\psi$ (NOT ψ) is a WFF too. The formula $\neg\psi$ asserts that ψ is false. Occurrences of domain variables in $\neg\psi$ are free or bound as they are free or bound in ψ.
(3) If ψ_1 and ψ_2 are WFF, then $\psi_1 \vee \psi_2$ (ψ_1 OR ψ_2) and $\psi_1 \wedge \psi_2$ (ψ_1 AND ψ_2) are also WFF asserting that "ψ_1 or ψ_2, or both, are true," and "ψ_1 and ψ_2 are both true," respectively. Occurrences of domain variables are free or bound in $\psi_1 \vee \psi_2$ or $\psi_1 \wedge \psi_2$ as they are free or bound in ψ_1 or ψ_2, depending on where they occur.
(4) If ψ_1 and ψ_2 are WFF, $\psi_1 \rightarrow \psi_2$ (IF ψ_1 THEN ψ_2), is also a WFF asserting "if ψ_1 is true then ψ_2 is true." Occurrences of domain variables are free or bound in $\psi_1 \rightarrow \psi_2$ as they are free or bound in ψ_1 or ψ_2, depending on where they occur.
(5) If ψ is a WFF and x is a domain variable free in ψ, then $\exists x(\psi)$ and $\forall x(\psi)$ are also WFF, with the occurrences of x being bound in both WFF. The formula $\exists x(\psi)$ asserts that a value of x exists such that when we substitute this value for all free occurrences of x in ψ, the formula ψ becomes true. The formula $\forall x(\psi)$ asserts that whatever the value of x, if we replace all free occurrences of x in ψ by this value, the formula ψ becomes true.
(6) If ψ is a WFF, then (ψ) is also a WFF.
(7) Nothing else is a WFF.

Parentheses may be placed around formulas as needed in order to change the precedence of the operator or to clarify the evaluation order. We assume that the order of precedence is: Atoms, quantifiers ($\exists$ and $\forall$), negation ($\neg$), conjunction ($\wedge$), disjunction ($\vee$), and implication ($\rightarrow$), in that order. Likewise, it is possible to introduce other operators such as XOR (eXclusive OR), NOR (NOT OR), NAND (NOT AND).... These have not been considered in order to avoid complication.

Note that we only use the existential and the universal quantifiers. However, it is possible to define other fuzzy quantifiers to relax the queries in some way. For example, we could use quantifiers like "the majority," "the minority," "approximately n," "approximately the middle".... The definition of this type of fuzzy quantifiers is a complex goal that we are currently investigating.

A WFF will be denoted $\psi(x_1, \ldots, x_n)$ when the domain variables $x_1, \ldots, x_n$ are free (not necessarily all existing in ψ). We will sometimes denote a WFF simply ψ, without this implying that ψ does not have any free variables.

If a constant k_i appears in the position i instead of the variable x_i, $\psi(x_1, \ldots, k_i, \ldots, x_n)$, we will suppose that all occurrences of the x_i variable may be substituted for k_i. Obviously, x_i is no longer a free variable, and therefore,

$$\psi(x_1, \ldots, x_{i-1}, k_i, x_{i+1}, \ldots, x_n) = \psi_1(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$$

where ψ_1 is the same WFF ψ but with all occurrences of x_i substituted for the constant k_i.

It is possible to simplify the former definition of WFF by removing the definition of some operators, as demonstrated in the following lemma:

LEMMA 1. *If ψ is a formula in domain relational calculus* (*or tuple calculus*, *for that matter*), *then there is an equivalent formula ψ' with no occurrences of $\wedge$, $\forall$, or $\rightarrow$.*

Proof. In order to remove the operators $\wedge$, $\forall$, and $\rightarrow$ in a formula, we substitute them with equivalent expressions in terms of the other operators:

(1) Substitute each subformula $\psi_1 \wedge \psi_2$ by $\neg(\neg \psi_1 \vee \neg \psi_2)$. This transformation is called DeMorgan's law.
(2) Substitute each subformula $\forall x(\psi(x))$ by $\neg \exists x(\neg \psi(x))$.
(3) Substitute each subformula $\psi_1 \rightarrow \psi_2$ by $\neg \psi_1 \vee \psi_2$.

Therefore, in ψ' only the operators $\neg$, $\vee$, or $\exists$ appear, which are the really essential ones. The other operators are useful to achieve both simpler and more intuitive expressions. ■

B. Restricting Relational Calculus to Yield Only Finite Relations

J. D. Ullman, in Ref. 14, defines the so-called safe expressions in classic relational calculus. The safe expressions are those which yield finite relations.

The nonsafe expressions must be ruled out since they are meaningless expressions. If we call a set of domain variables $x_1, \dots, x_n$ by the name X, an example of a nonsafe expression is $\{X \mid \neg R(X)\}$, which denotes all possible tuples that are not in R, something which is impossible to retrieve if any domain of R is infinite.

To define safety, Ullman defines DOM(ψ) as the set of symbols that either appear explicitly in expression ψ or are components of some tuple in some relation mentioned in ψ. Ullman therefore says that an expression of classic relational calculus $\{X \mid \psi(X)\}$ is safe if:

(1) Whenever X satisfies ψ, all components of X are members of DOM(ψ).
(2) For each subformula of ψ of the form $\exists u(w(u))$, if u satisfies w for any values of the other free variables in w, then $u \in$ DOM(w).
(3) For each subformula of ψ of the form $\forall u(w(u))$, if $u \notin$ DOM(w), then u satisfies w for all values of the other free variables in w.

As Ullman says, the purpose of points (2) and (3) is to assure that we can determine the truth of a quantified formula (with $\exists$ or $\forall$) by considering only those u values belonging to DOM(w). For example, any formula,

$$\exists u(R(u, \dots) \wedge \cdots)$$

satisfies (2), and any formula,

$$\forall u(\neg R(u, \dots) \vee \cdots)$$

satisfies (3). Note that in the definition of safety, we do not assume that any free variables of w, besides u, necessarily have values in DOM(w). Rules (2) and (3) must remain independent of the value of those variables.

All variables in a WFF must, at the very least, form part of a fuzzy ownership atom. So, a variable has the domain of the corresponding attribute according to the position of this variable in the atom. This variable can only take the values of that attribute in the relation (or relations). In some versions of relational calculus (like QUEL[4]), it is possible to previously establish the range of each variable, such that the formula is, in some cases, lightly simplified.

In fuzzy relational calculus, due to the fuzzy characteristics of its fuzzy context and due to the fact that we can establish fulfilment degrees, it can sometimes be useful to use nonsafe expressions in the classic sense and handle them as if they were safe, i.e., handle them via the so-called "limited evaluation," restricting their evaluation only with values in DOM(ψ).

DEFINITION 3.1. *In fuzzy relational calculus we will say that an expression is safe, if on establishing all its thresholds to* 1, *the expression is safe when considered in the classic sense. All expressions will be evaluated via the limited evaluation, as if they were safe: in order to compute the query results it will be obligatory that all the values in the result belong to* DOM(ψ), *i.e., the result is always restricted exclusively to values in* DOM(ψ).

The previous definition is logical because to consider all possible tuples in our universe is meaningless (there can be an infinite number of tuples). Then, as in classic calculus, we only deal with those tuples which exist in the relevant relations of our database. In the following section we study some cases in which this is clarified.

IV. EXPRESSIVE POWER OF THE FUZZY DOMAIN RELATIONAL CALCULUS

For every expression of fuzzy relational algebra, there is an equivalent expression in relational calculus. This affirmation is fundamental for relational completeness and it is demonstrated below (Theorem 1).

In this section, we will first express the relational primitive algebraic operators in fuzzy domain relational calculus (union, difference, Cartesian product, projection, and selection). We will then express other nonprimitive algebraic operators in calculus (intersection, quotient, and join).

The queries in fuzzy calculus are especially expressive due to two main reasons: the calculus is nonprocedural and therefore we must only indicate what we want (not how to obtain it), and, moreover, we can establish the thresholds (γ) which control the minimum fulfilment degree that the values in the resulting tuples have.

In classic databases, the fulfilment degree must always be maximum, a goal achieved with $\gamma = 1$ in all fuzzy atoms. This does not necessarily indicate that the matching between all different fuzzy values must be accurate, but that a total possibility exists of both values being (or referring to) the same value. Naturally, if all possible fuzzy values in the domain are well-defined then with $\gamma = 1$ we ensure accurate matching between the compared values. With "well-defined fuzzy values" we indicate the (x, y) values such that, if they are considered different they are not as similar as those in which $\Theta^{=}(x, y) = 1$ [see Eq. (8)].

Note that in some expressions, to put a threshold $\gamma < 1$ is meaningless in ownership atoms because we apply the limited evaluation. We will later explain this better using the union expression as an example. We will explicitly indicate those atoms in which it may be useful to put a threshold of $\gamma < 1$.

A. Primitive Algebraic Operators

Below, we express in fuzzy domain relational calculus the fuzzy primitive algebraic operators:

(1) Fuzzy Union: Let R and R' be two relations of arity n, compatible with regards to the union (with the same number and type of attributes). The union of both relations is given by the following expression in calculus,

$$R \cup R' = \{x_1, \ldots, x_n \mid R(x_1, \ldots, x_n) \vee R'(x_1, \ldots, x_n)\} \qquad (11)$$

Put into words, this expression yields the set of tuples $(x_1, \ldots, x_n)$ such that they are in R or in R'. See Example 5 in Section VI.

It is useless to put a threshold $\gamma < 1$ on the ownership atoms in the union expression. Each resulting tuple will belong with a degree of 1 to one of the two relations: R or R'. The ownership degree to the other relation is not important. This expression is safe in a classic sense and therefore we can say it is safe in our fuzzy relational calculus. However, in a fuzzy sense that expression is not safe because there may be tuples satisfying ψ with all or some of the elements outside DOM(ψ). The smaller the thresholds are, the more frequently this should occur. In this case, the result is restricted to values in DOM(ψ). The fulfilment thresholds lose their meaning and so they are not put in the expression.

(2) Fuzzy Difference: Let R and R' be two relations of arity n, compatible with regards to the union. The difference between both relations is given by

$$R - R' = \{x_1, \dots, x_n \mid R(x_1, \dots, x_n) \wedge \neg R'(x_1, \dots, x_n) \geq \gamma\} \tag{12}$$

Put into words, this expression yields the set of tuples $(x_1, \dots, x_n)$ such that they are in R and not in R' with a degree greater than or equal to γ. With $\gamma = 1$, we have a difference similar to the classic style. With $\gamma < 1$, tuples in R that belong to R' with a sufficiently large degree (γ) are also removed in the resulting relation. This ownership to R' is calculated by Eq. (7). See Example 12 in Section VI.

(3) Fuzzy Cartesian Product (or Times): Let R and R' be two relations of arity n and m, respectively. The Cartesian product between both relations is expressed by

$$R \times R' = \{x_1, \dots, x_n, v_1, \dots, v_m \mid R(x_1, \dots, x_n) \wedge R'(v_1, \dots, v_m)\} \tag{13}$$

Put into words, this expression yields the set of all possible tuples $(x_1, \dots, x_n, v_1, \dots, v_m)$ such that $(x_1, \dots, x_n)$ belong to R and $(v_1, \dots, v_m)$ belong to R'.

(4) Fuzzy Projection: Let R be a relation of arity n and $(A_1, \dots, A_k)$ be a set of R attributes with $k < n$. Then, the projection of R onto these attributes is expressed by

$$\mathscr{P}(R; A_1, \dots, A_k) = \{x_1, \dots, x_k \mid \exists x_{k+1}, \dots, x_n \; R(x_1, \dots, x_n)\} \tag{14}$$

In order to simplify the expression, we suppose that the attributes onto which the projection is made are the first k attributes. The extrapolation in the case of them not being the first ones is trivial. This expression yields a relation similar to R, but removing those attributes which are not projected onto. See Examples 11 and 12 in Section VI.

(5) Fuzzy Selection: Let R be a relation of arity n and $\mathscr{F}$ a formula expressing a condition that tuples in R must satisfy. Then, the selection on R with the condition $\mathscr{F}$ is expressed by

$$\mathscr{S}(R; \mathscr{F}) = \{x_1, \dots, x_n \mid R(x_1, \dots, x_n) \wedge \mathscr{F}'\} \tag{15}$$

where $\mathscr{F}'$ is the formula $\mathscr{F}$ with each operand i, denoting the ith component, replaced by x_i. This expression yields a relation with tuples in R satisfying the predicate $\mathscr{F}$ (or the WFF $\mathscr{F}'$). See Example 11 in Section VI.

B. Nonprimitive Algebraic Operators

There are other very useful operators in relational algebra which are not primitive, i.e., they may be expressed in terms of primitive operators. We will express the most typical operators in fuzzy relational calculus:

(1) Fuzzy Intersection: Let R and R' be two relations of arity n, compatible with regards to the union. The intersection of both relations is given by the following calculus expression,

$$R \cap R' = \{x_1, \ldots, x_n \mid R(x_1, \ldots, x_n) \geq \gamma \wedge R'(x_1, \ldots, x_n) \geq \gamma'\} \quad (16)$$

Put into words, this expression yields the set of tuples $(x_1, \ldots, x_n)$ such that they belong to R (with a minimum degree of γ) and they belong to R' (with a minimum degree of γ'). If we observe this expression we can see that it is possible that tuples exist which belong to R and R' with a degree grater than or equal to γ and γ', respectively, and with values outside DOM(ψ). In order to compute the result we apply limited evaluation restricting it to tuples with values in DOM(ψ). With this restriction, the fulfilment threshold γ of the atom of R will be useful in the R' tuples (because R tuples belong to R with a degree of 1). Likewise, the threshold γ' is applied to$R$ tuples. So, an R tuple belongs to the intersection if it belongs to R' with a minimum degree of γ'. If $\gamma = \gamma' = 1$ we have an intersection similar to the classic style. See Example 6 in Section VI.

(2) Fuzzy Quotient (or Division): The definition of the division operator in relational algebra is:

DEFINITION 4.1. *Let R and R' be relations with headers (A, B) and (B), respectively, where A and B are simple attributes or sets of attributes. Then, the relational quotient of R by R', denoted by $R \div R'$, is a relation Q with header (A) whose body is formed by all the tuples $(A\colon a)$ so that a tuple $(A\colon a, B\colon b)$ exists in R for every tuple $(B\colon b)$ in R'.*

Let R and R' be two relations of arity $n + m$ and m, respectively, with the R' attributes being of the same type as the last m attributes of R, defined by

$$R(a_1, \ldots, a_n, b_1, \ldots, b_m)$$
$$R'(b_1, \ldots, b_m)$$

Then, the relational quotient, $R \div R'$, is expressed in the following query: *Give me the tuples $(a_1, \ldots, a_n)$ in R which are related (in R) with all the tuples in R' (with a minimum degree of γ).* In fuzzy databases the relation is by similarity (not by equality). Therefore, we can give a fulfilment degree of γ to this similarity.

In fuzzy domain relational calculus this query is obtained by the expression,

$$R \div R' = \{a_1, \ldots, a_n \mid \forall b_1, \ldots, b_m (R'(b_1, \ldots, b_m) \rightarrow R(a_1, \ldots, a_n, b_1, \ldots, b_m) \geq \gamma)\} \quad (17)$$

This expression yields the set of tuples A such that if B belongs to R' then (A, B) belongs to$R$ with a minimum degree of γ.

Note that the atom of R has a fulfilment threshold of γ. This makes is possible for there to be tuples in the result which partially satisfy the query condition. If we establish that $\gamma = 1$ we will only retrieve the R tuples $(a_1, \dots, a_n)$ which are *exactly* related (in R) with all the values of R'. Thus, if $\gamma = 1$ then we have a division similar to the classic style. When $\gamma < 1$ we also retrieve those tuples which are similar to all the R' tuples of minimum degree γ. See Example 11 in Section VI.

Applying Lemma 1 to expression (17) we obtain the equivalent of

$$\{a_1, \dots, a_n \mid \neg \exists b_1, \dots, b_m \neg(\neg R'(b_1, \dots, b_m) \vee R(a_1, \dots, a_n, b_1, \dots, b_m) \geq \gamma)\}$$

This expression is simplified by applying DeMorgan's law,

$$\{a_1, \dots, a_n \mid \neg \exists b_1, \dots, b_m (R'(b_1, \dots, b_m) \wedge \neg R(a_1, \dots, a_n, b_1, \dots, b_m) \geq \gamma)\}$$

(3) Fuzzy Join: Let R and R' be two relations of arity n and m. The join of both relations is expressed by

$$R \underset{\Theta^{\theta}(x_i, v_j) \geq \gamma}{\bowtie} R' = \{x_1, \dots, x_n, v_1, \dots, v_m \mid R(x_1, \dots, x_n) \wedge R'(v_1, \dots, v_m) \wedge \Theta^{\theta}(x_i, v_j) \geq \gamma\} \quad (18)$$

with $i \in (1, \dots, n\}$ and $j \in \{1, \dots, m\}$. Put into words, this expression yields the set of tuples $(x_1, \dots, x_n, v_1, \dots, v_m)$ such that $(x_1, \dots, x_n)$ belong to R, $(v_1, \dots, v_m)$ belong to R' and they satisfy the condition,

$$\Theta^{\theta}(x_i, v_j) \geq \gamma$$

x_i and v_j being two variables in R and R', respectively. Note that a join is a selection on the Cartesian product.

The natural join is a join in which all attributes with the same name in both relations are compared using the fuzzy comparator approximately equal [Eq. (8)]. One of the two compared attributes is removed from the result. In a fuzzy natural joint it may be interesting not to remove those attributes or to fuse, in some way, the two attributes into a single attribute.

C. Reduction of Fuzzy Relational Algebra to Fuzzy Domain Relational Calculus

THEOREM 1. *For any expression E in fuzzy relational algebra there is an equivalent safe expression in fuzzy domain relational calculus.*

Proof. Using the above expressions, the proof proceeds by induction over the number of operators in E: if E has no operators then it is a relation R without operations. In this case, if R has arity n the domain calculus expression is

$$R = \{x_1, \dots, x_n \mid R(x_1, \dots, x_n)\}$$

If E has one or more operators, the procedure will be in accordance with the main operator. The main operator may be the union ($E = E_1 \cup E_2$), the difference ($E = E_1 - E_2$), the Cartesian product ($E = E_1 \times E_2$), the projection ($E = \mathscr{P}(E_1; A_1, \ldots, A_k)$), or the selection ($E = \mathscr{S}(E_1; \mathscr{F})$). In each case, Eq. (11)–(15) is applied, respectively, replacing the ownership atoms R and R' in the equations by the WFF of the calculus expressions of E_1 and E_2, respectively. ■

V. COMPUTING THE RESULTING GENERALIZED FUZZY RELATION

With all the former definitions, we can write any expression in this fuzzy domain relational calculus. However, in fuzzy databases it is very useful to know the fulfilment degree to which a concrete value satisfies a condition. The GEFRED model has the so-called compatibility attributes, the C_j of each A_j attribute in each generalized fuzzy relation (see Definition 2.2). In order to compute the values of C_j, the so-called c_{ij} "compatibility degrees" of each attribute in a tuple, we will define the degree of a domain variable in a WFF:

A. Degree of a Variable in a Well Formed Formula with a Substitution

We will define a function Δ which will be applied to a WFF ψ. This function will return the degree to which a concrete value in a concrete tuple satisfies the predicate ψ. The function needs the value (x) that we want to evaluate and its tuple (S): Δ_x^S. This computation does not use the fulfilment degrees, γ, of the atoms of ψ.

DEFINITION 5.1. *Let $\mathscr{E}$ be the set of all safe expressions of fuzzy domain relational calculus, Ψ be the set of all* WFF *in $\mathscr{E}$, $\psi(x_1, \ldots, x_n) \in \Psi$ be* WFF *with all the x_i as unique free variables, and x be a domain variable (which either appears or not in ψ). Then, to evaluate if ψ is True or False we must substitute the free variables in ψ by values $(s_1, \ldots, s_n)$ which we will call substitution S.*

We suppose, by Lemma 1, that only three types of basic operators exist in ψ: negation ($\neg$), disjunction ($\vee$), and existential quantifier ($\exists$).

We then recursively define the so-called degree of a domain variable x in a WFF *ψ with a substitution S, denoted by $\Delta_x^S(\psi)$, as a function,*

$$\begin{gathered} \Delta_x^S \colon \Psi \to [0,1] \cup \lambda \\ \Delta_x^S(\psi) \in [0,1] \cup \lambda \end{gathered} \tag{19}$$

where λ is a constant with the requisite $\lambda \notin [0,1]$. This constant indicates that the degree of the variable x in ψ is not applicable or meaningless. We set $\lambda < 0$ (in order to simplify the disjunction definition).

Depending on the type of the main operator in ψ, four cases exist:

(1) *Zero operators*: *In this case ψ is a fuzzy atom and it will distinguish the two types of fuzzy atoms*: *Ownership and Comparison.*
(2) *Negation.*
(3) *Disjunction.*
(4) *Existential Quantifier.*

We study these four cases in order to compute the value of $\Delta_x^S(\psi)$. The definition is recursive, where the base case is case (10 (*when ψ is an atom*). *In all other cases, the degree in ψ is computed starting from the degrees in the subformulas obtained by evaluating the main operator in ψ*:

(1) *Zero operators* (ψ *is an atom*): *In this case, there are two types of atoms*:
 (a) *Ownership*:

$$\Delta_x^S(R(x_1, \ldots, x_n, K) \geq \gamma) = \begin{cases} R(K) & \textit{if there are no variables in } \psi \\ R(S, K) & \textit{if } x = A_i \textit{ and } \nexists\, C_i \\ \min\{c_{ri}, R(S, K)\} & \textit{if } x = A_i \textit{ and } \exists C_i \\ \lambda & \textit{in any other case} \end{cases} \tag{20}$$

where:

- *R is a generalized fuzzy relation of arity $n + p$.*
- *K is a list with all constant $(k_1, \ldots, k_p)$ in the atom.*
- *The values of $R(S, K) = R(s_1, \ldots, s_n, k_1, \ldots, k_p)$ and $R(K)$ are computed by Eq.* (7).
- *A_i with $i \in \{1, \ldots, n\}$, is an attribute of R. The comparison $x = A_i$ indicates that the variable x has the domain of the A_i attribute.*
- *c_{ri} is the value of the compatibility attribute C_i in R, associated with the A_i attribute, in a tuple r such that its value component $(\tilde{d}_{r1}, \ldots, \tilde{d}_{rn})$ is the most similar tuple to (S, K) in R. The most similar tuple to (S, K) in a relation always exists and it is even possible that some tuples exist with the same similarity. In the case of some tuples existing, the greatest c_{ri} will be taken.*

Computing the degree Δ_x^S in an ownership atom $R(x_1, \ldots, x_n, K)$ is easy when the following algorithm is used:

IF $\nexists$ *variables in the atom*
  RETURN $R(K)$
ELSE
  IF $x = A_i$ *with* $i \in \{1, \ldots, n + p\}$ THEN (* If x is an R attribute *)
    IF $\exists C_i$ *in* R THEN
      Search the rth *tuple in* R *such that it is the*
      most similar tuple *to* (S, K) *in* R *with the greatest* c_{ri}:
      RETURN $\min\{c_{ri}, R(S, K)\}$
    ELSE (* $\nexists\, C_i$ *)
      RETURN $R(S, K)$
    END IF
  ELSE (* If x is NOT an attribute of R *)
    RETURN λ
  END IF
END IF

The following algorithm is used to "Search the rth tuple in R such that it is the most similar tuple to (S, K) in R with the greatest c_{ri}" *and* RETURN *the relevant result. We suppose R has m tuples and we rename tuple (S, K) as $Y = (y_1, \dots, y_{n+p})$:*

```
G := 0 (*Greatest similarity: R(S, K)*)
C := 0 (*Greatest compatibility degree: c_ri *)
FOR t := 1 TO m DO (*For each tuple in R... *)
  M := min_{w=1,...,n+p}{Θ⁺(d̃_tw, y_w)}
  (*M := Similarity between Y and the tth tuple*)
  IF G < M THEN
    G := M
    C := c_ti
  ELSE
    IF G = M THEN
      IF C < c_ti THEN
        C := c_ti
      END IF
    END IF
  END IF
END FOR
RETURN min{C, G}
```

(b) *Comparison*:

$$\Delta_x^S\big(\Theta^\theta(x_i, y) \geq \gamma\big) = \begin{cases} \Theta^\theta(s_i, y) & \textit{if } x_i \textit{ is a variable and } x = x_i \\ \Theta^\theta(x_i, y) & \textit{if } x_i \textit{ is a constant} \\ \lambda & \textit{in any other case} \end{cases} \tag{21}$$

where s_i is the value of S corresponding to the variable x_i. The equality $x = x_i$ indicates that both variables are the same variable (with the same domain).

(2) *Negation*: $\psi(x_1, \dots, x_n) = \neg\, \psi_1(x_1, \dots, x_n)$,

$$\Delta_x^S(\psi(x_1, \dots, x_n)) = \begin{cases} 1 - \Delta_x^S(\psi_1(x_1, \dots, x_n)) & \textit{if } \Delta_x^S(\psi_1(x_1, \dots, x_n)) \neq \lambda \\ \lambda & \textit{in any other case} \end{cases} \tag{22}$$

(3) *Disjunction*: $\psi(x_1, \dots, x_n) = \psi_1(u_1, \dots, u_p) \vee \psi_2(v_1, \dots, v_q)$, *where each u_j is a distinct x_k and each v_j is a distinct x_k, although some of the us and vs may be the same x_k. Thus,*

$$\Delta_x^S(\psi(x_1, \dots, x_n)) = \max\big\{\Delta_x^S(\psi_1(u_1, \dots, x_p)), \Delta_x^S(\psi_2(v_1, \dots, v_q))\big\} \tag{23}$$

As $\lambda < 0$, we are certain that the disjunction degree is λ only if the degrees of both disjunction parts are λ.

(4) *Existential Quantifier*: $\psi(x_1, \dots, x_n) = \exists x_{n+1}(\psi_1(x_1, \dots, x_{n+1}))$,

$$\Delta_x^S(\psi(x_1, \dots, x_n)) = \max_{s_{n+1} \in \mathrm{DOM}(\psi)} \big\{\Delta_x^S(\psi_1(x_1, \dots, x_n, s_{n+1}))\big\} \tag{24}$$

Note that $\mathrm{DOM}(\psi) = \mathrm{DOM}(\psi_1)$. *We will call ψ_1 a subformula of $\exists$. On evaluating the degree of ψ_1, s_{n+1} will be a constant and it will be treated as such. Naturally, the value of s_{n+1} will be within the domain of the variable x_{n+1}.*

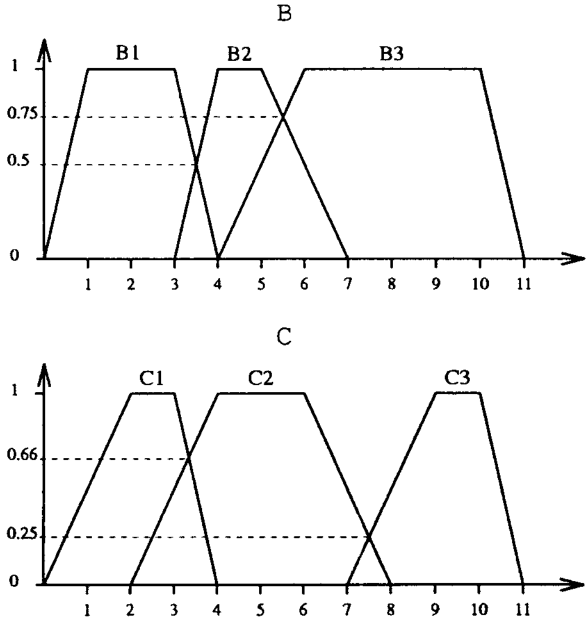

Figure 1. Definition of labels on attributes B and C (Example 2).

Example 2. Let us suppose a relation R where its two attributes are B and C. The linguistic labels in Figure 1 are defined on them. Furthermore, let us suppose that there are not yet any compatibility attributes in R.

Let ψ be the following WFF,

$$\psi(b,c) = R(b,c) \wedge (\Theta^{=}(b,B2) \geq 0.7 \rightarrow \Theta^{=}(c,C2) \geq 0.5)$$

where $\Theta^{=}$ is the generalized fuzzy comparator in Eq. (8).

If we have a substitution $S = (B3, C1) \in R$, we can compute $\Delta_c^S(\psi(b,c))$. Then we apply Lemma 1 to ψ, obtaining

$$\psi(b,c) = \neg(\neg R(b,c) \vee \neg(\neg \Theta^{+}(b,B2) \geq 0.7 \vee \Theta = (c,C2) \geq 0.5))$$

A very easy way to compute $\Delta_c^S(\psi(b,c))$ is in the following three steps:

(1) In ψ substitute all its atoms by the degrees of the same variable c in each atom with the same substitution S,

$$\Delta_c^S(\psi(b,c)) \equiv \neg\big(\neg \Delta_c^S(R(b,c)) \vee \neg\big(\neg \Delta_c^S(\Theta^{=}(b,B2) \geq 0.7) \vee \Delta_c^S(\Theta^{=}(c,C2) \geq 0.5)\big)\big)$$

(2) Compute each degree in each atom independently and replace it in its place,

$$\Delta_c^S(\psi(b,c)) \equiv \neg(\neg 1 \vee \neg(\neg \lambda \vee 0.66))$$

(3) Operate according to Definition 5.1. In general, the values different from λ remain and the λ values are ruled out. The operators are evaluated from the greatest to the smallest precedence,

$$\begin{aligned}
\Delta_c^S(\psi(b,c)) &\equiv \neg(1-1 \vee \neg(\lambda \vee 0.66)) \\
&\equiv \neg(0 \vee \neg(0.66)) \\
&\equiv \neg(0 \vee (1-0.66)) \\
&\equiv \neg(0 \vee 0.34) \\
&\equiv \neg(0.34) \\
&\equiv 1-0.34 \\
&\equiv 0.66
\end{aligned}$$

Likewise we can compute the degree of b in ψ with the substitution S,

$$\begin{aligned}
\Delta_b^S(\psi(b,c)) \equiv \neg\big(\neg\,\Delta_b^S(R(b,c)) \vee \neg\big(\neg\,\Delta_b^S(\Theta^{=}(b,B2) \geq 0.7) \\
\vee\,\Delta_b^S(\Theta^{=}(c,C2) \geq 0.5)\big)\big)
\end{aligned}$$

$$\begin{aligned}
&\equiv \neg(\neg 1 \vee \neg(\neg 0.75 \vee \lambda)) \\
&\equiv \neg(1-1 \vee \neg(1-0.75 \vee \lambda)) \\
&\equiv \neg(0 \vee \neg(0.25)) \\
&\equiv \neg(\neg(0.25)) \\
&\equiv 0.25
\end{aligned}$$

The constant λ is a symbol indicating that we must remove that part and center the computation in the other part. ■

Looking at the former example, we can build the following lemma in order to simplify some operations where a conjunction operator exists:

LEMMA 2. *Let ψ be a* WFF *whose main operator is $\wedge$ (conjunction), i.e., the formula is $\psi(x_1,\ldots,c_n) = \psi_1(u_1,\ldots,u_p) \wedge \psi_2(v_1,\ldots,v_q)$, where each u_j is a distinct x_k and each v_j is a distinct x_k, although some of the us and vs may be the same x_k. In this case,*

$$\Delta_x^S(\psi(x_1,\ldots,x_n)) = \begin{cases} \Delta_x^S(\psi_1(u_1,\ldots,x_p)) \\ \quad \textit{if } \Delta_x^S(\psi_1) \neq \lambda \textit{ and } \Delta_x^s(\psi_2) = \lambda \\ \Delta_x^S(\psi_2(v_1,\ldots,v_q)) \\ \quad \textit{if } \Delta_x^S(\psi_1) = \lambda \textit{ and } \Delta_x^S(\psi_2) \neq \lambda \\ \min\{\Delta_x^S(\psi_1(u_1,\ldots,x_p)), \Delta_x^S(\psi_2(v_1,\ldots,v_q))\} \\ \quad \textit{in any other case} \end{cases} \tag{25}$$

Proof. By Lemma 1 (DeMorgan's law) we obtain

$$\psi(x_1,\dots,x_n) = \neg\big(\neg\,\psi_1(u_1,\dots,u_p) \vee \neg\,\psi_2(v_1,\dots,v_q)\big)$$

Let α and β be

$$\alpha = \Delta_x^S\big(\psi_1(u_1,\dots,x_p)\big)$$

$$\beta = \Delta_x^S\big(\psi_2(v_1,\dots,v_q)\big)$$

There are four distinct cases:

(1) $\alpha \neq \lambda$ and $\beta = \lambda$: In this case the obtained result according to this lemma is α. Applying Definition 5.1 we also obtain the same result,

$$\Delta_x^S(\psi(x_1,\dots,x_n)) \equiv \neg(\neg\,\alpha \vee \neg\,\lambda) \equiv \neg(1-\alpha \vee \lambda)$$
$$\equiv \neg(1-\alpha) \equiv 1-(1-\alpha) = \alpha$$

(2) $\alpha = \lambda$ and $\beta \neq \lambda$: Here the result obtained according to this lemma is β. Applying Definition 5.1, we also obtain β,

$$\Delta_x^S(\psi(x_1,\dots,x_n)) \equiv \neg(\neg\,\lambda \vee \neg\,\beta) \equiv \neg(\lambda \vee 1-\beta)$$
$$\equiv \neg(1-\beta) \equiv 1-(1-\beta) = \beta$$

(3) $\alpha \neq \lambda$ and $\beta \neq \lambda$: The result is $\min\{\alpha,\beta\}$, the same as that which is obtained by Definition 5.1,

$$\Delta_x^S(\psi(x_1,\dots,x_n)) \equiv \neg(\neg\,\alpha \vee \neg\,\beta) \equiv 1-\max\{1-\alpha, 1-\beta\}$$

It is necessary to take into account that: $\min\{\alpha,\beta\} = 1-\max\{1-\alpha,1-\beta\}$.

(4) $\alpha=\lambda$ and $\beta=\lambda$: Here the result obtained according to this lemma is $\min\{\lambda,\lambda\} = \lambda$. Applying Definition 5.1, we obtain the same result,

$$\Delta_x^S(\psi(x_1,\dots,x_n)) \equiv \neg(\neg\,\lambda \vee \neg\,\lambda) \equiv \neg(\lambda \vee \lambda) \equiv \neg\,\lambda \equiv \lambda$$

If we establish that $\lambda > 1$, Eq. (25) is simplified by only taking into account the third case, but Eq. (23) of the disjunction will not be so simple. ■

We present two other lemmas which simplify the computation of function Δ when implication ($\rightarrow$) or universal quantifier ($\forall$) operators exist.

LEMMA 3. *Let ψ be a* WFF *whose main operator is $\rightarrow$ (implication), i.e., the formula is $\psi(x_1,\dots,x_n) = \psi_1(u_1,\dots,u_p) \rightarrow \psi_2(v_1,\dots,v_q)$, where each u_j is a distinct x_k and each v_j is a distinct x_k, although some of the us and vs may be the same x_k. In this case,*

$$\Delta_x^S(\psi(x_1,\dots,x_n))$$
$$= \begin{cases} 1-\Delta_x^S\big(\psi_1(u_1,\dots,x_p)\big) & \text{if } \Delta_x^S(\psi_1) \neq \lambda \text{ and } \Delta_x^S(\psi_2) = \lambda \\ \Delta_x^S\big(\psi_2(v_1,\dots,v_q)\big) & \text{if } \Delta_x^S(\psi_1) = \lambda \text{ and } \Delta_x^S(\psi_2) \neq \lambda \\ \max\{1-\Delta_x^S(\psi_1), \Delta_x^S(\psi_2)\} & \text{if } \Delta_x^S(\psi_1) \neq \lambda \text{ and } \Delta_x^S(\psi_2) \neq \lambda \\ \lambda & \text{if } \Delta_x^S(\psi_1) = \lambda \text{ and } \Delta_x^S(\psi_2) = \lambda \end{cases} \qquad (26)$$

Proof. By Lemma 1 we obtain

$$\psi(x_1,\dots,x_n) = \neg\,\psi_1(u_1,\dots,u_p) \vee \psi_2(v_1,\dots,v_q)$$

Let α and β be

$$\alpha = \Delta_x^S(\psi_1(u_1,\dots,x_p))$$

$$\beta = \Delta_x^S(\psi_2(v_1,\dots,v_q))$$

There are four distinct cases:

(1) $\alpha \neq \lambda$ and $\beta = \lambda$: In this case the result obtained according to this lemma is $1-\alpha$. By Definition 5.1 we also obtain the same result,

$$\Delta_x^S(\psi(x_1,\dots,x_n)) \equiv \neg\,\alpha \vee \lambda \equiv (1-\alpha) \vee \lambda \equiv 1-\alpha$$

(2) $\alpha = \lambda$ and $\beta \neq \lambda$: In this case the result obtained is β. By Definition 5.1 we also obtain β,

$$\Delta_x^S(\psi(x_1,\dots,x_n)) \equiv \neg\,\lambda \vee \beta \equiv \lambda \vee \beta \equiv \beta$$

(3) $\alpha \neq \lambda$ and $\beta \neq \lambda$: The result is now $\max\{1-\alpha, \beta\}$, the same as that which is obtained by Definition 5.1,

$$\Delta_x^S(\psi(x_1,\dots,x_n)) \equiv \neg\,\alpha \vee \beta \equiv (1-\alpha) \vee \beta \equiv \max\{1-\alpha,\beta\}$$

(4) $\alpha = \lambda$ and $\beta = \lambda$: By this lemma we obtain λ, the same as that which is obtained by Definition 5.1,

$$\Delta_x^S(\psi(x_1,\dots,x_n)) \equiv \neg\,\lambda \vee \lambda \equiv \lambda \vee \lambda \equiv \lambda$$

■

LEMMA 4. *Let ψ be a* WFF *whose main operator is* $\forall$ (*universal quantifier*), *i.e., the formula is* $\psi(x_1,\dots,x_n) = \forall x_{n+1}\eta_1(x_1,\dots,x_{n+1})$. *then*

$$\Delta_x^S(\psi(x_1,\dots,x_n)) = \min_{s_{n+1}\in \mathrm{DOM}(\psi)} \{\Delta_x^S(\psi_1(x_1,\dots,x_n,s_{n+1}))\} \tag{27}$$

We will call ψ the subformula of $\forall$ to ψ_i. Evaluating the degree of ψ_1, s_{n+1} will be a constant and it will be treated as such.

Proof. By Lemma 1 we have

$$\psi(x_1,\dots,x_n) = \neg\,\exists x_{n+1} \neg\,\psi_1(x_1,\dots,x_{n+1})$$

Let $(\alpha_1,\dots,\alpha_f)$ be all the f values of s_{n+1}, and

$$\beta_i = \Delta_x^S(\psi_1(x_1,\dots,x_n,\alpha_i))$$

with $i = 1,\dots,f$. Then, according to this lemma the result obtained is

$$\Delta_x^S(\psi(x_1,\dots,x_n)) = \min\{\beta_1,\dots,\beta_f\}$$

By Definition 5.1, the result obtained is

$$\Delta_x^S(\psi(x_1,\ldots,x_n)) \equiv \neg\, \Delta_x^S(\exists x_{n+1} \neg\, \psi_1(x_1,\ldots,x_{n+1}))$$
$$\equiv 1 - \max\{1-\beta_1,\ldots,1-\beta_f\}$$

We can see that both results are equivalent. ■

B. Resulting Generalized Fuzzy Relation

With Definition 5.1 we can now compute the generalized fuzzy relation resulting from a safe expression in fuzzy domain relational calculus in the form,

$$\{x_1, x_2, \ldots, x_n \mid \psi(x_1, x_2, \ldots, x_n)\}$$

The result of this expression is a generalized fuzzy relation R as follows,

$$R = \begin{cases} \mathscr{H} = \{(A_1\colon D_1[,C_1]),\ldots,(A_n\colon D_n[,C_n])\} \\ \mathscr{B} = \left\{(A_1\colon \tilde{d}_{r1}[,c_{r1}]),\ldots,(A_n\colon \tilde{d}_{rn}[,c_{rn}])\right\} \end{cases}$$

with $r = 1,2,\ldots,m$, m being the number of tuples of the relation. Then, in order to retrieve $\mathscr{H}$ and $\mathscr{B}$, we carry out two steps:

- First, we compute the value component of the body, $\{(A_1\colon \tilde{d}_{r1}),\ldots,(A_n\colon \tilde{d}_{rn})\}$, formed by all tuples $(\tilde{d}_{r1},\ldots,\tilde{d}_{rn})$ that satisfy (make True) the predicate $\psi(\tilde{d}_{r1},\ldots,\tilde{d}_{rn})$. We call the number of tuples that satisfy the predicate m.
- The compatibility component of the body, $\{[c_{r1}],\ldots,[,c_{rn}]\}$, is computed after, taking into account that in the head, the compatibility attribute C_i, with $i \in [1,n]$, exists if and only if $\Delta_{x_i}^{S_r}(\psi(x_1,\ldots,x_n)) \neq \lambda$, for all the substitutions $S_r = (\tilde{d}_{r1},\ldots,\tilde{d}_{rn})$ with $r = 1,2,\ldots,m$. It is easy to observe that if the degree is equal to λ for one substitution, then it will be equal to λ for all the rest of the substitutions. Then, if attribute C_i exists, its compatibility degrees, for all the m tuples are computed in the following way,

$$c_{ri} = \Delta_{x_i}^{S_r}(\psi(x_1,\ldots,x_n)) \tag{28}$$

  where $r = 1,2,\ldots,m$ and $S_r = (\tilde{d}_{r1},\ldots,\tilde{d}_{rn})$ is the tuple rth of R. Also, we may consider that $\nexists\, C_i$ (and so it can be removed) if $c_{ri} = 1\ \forall r = 1,\ldots,m$. If all tuples have a value 1 in a concrete compatibility attribute then this compatibility attribute does not give us any information, because this is what we suppose if the compatibility attribute does not exist.

We apply function Δ with substitutions s_r, i.e., with all the tuples that satisfied the query predicate. This could pose the question of how to obtain these tuples so that afterward the corresponding function Δ can be applied to them. However, this question is meaningless because we are defining a relational calculus and this is, by definition, a nonprocedural language, i.e., its expressions say what we want to retrieve but do not say how to retrieve it. Once the tuples of result (R) have been obtained, in whatever way, the compatibility degree of each value of each tuple is then computed, using function Δ.

VI. EXAMPLES

The fuzzy relational calculus has a greater expressive power than classic relational calculus. In order to demonstrate this power and to clarify the computation of the resulting generalized fuzzy relation in a query, some examples are given in this section which include the vase casuistry that may be found in a query. Each of the following examples focuses on one or several of these particular cases, but all are based in the same context which is explained below.

Supposing we have a fuzzy relational database of basketball players. A database relation may have the attributes (PLAYER, TEAM, HEIGHT, QUALITY, NUM_SHIRT...). We will use a projection of this relation shown in Table III. The fields HEIGHT (where the player's height is stored) and QUALITY (where the player's quality is measured according to his average points per match) allow fuzzy values (type 6 in Table I). For the sake of the examples, we will use the linguistic labels in Figure 2.

We have eliminated the labels "Very short" and "Very bad," since in our opinion professional players with these characteristics do not exist.

We use the initial letter of the name of the attribute to whose domain they belong $(p, t, h,$ and $q)$ as identifiers of the domain variables. We will call the predicates ψ, as above. We will call the resulting relation of every example R_i, where i is the example number.

Example 3. Show the players with their teams and the heights of those *Short* players (with a minimum degree of 0.5).

Table III. Relation R.

$\mathscr{H}$	PLAYER	TEAM	HEIGHT	QUALITY
$\mathscr{B}$	J1	Almería	Tall	Very good
	J2	Almería	Short	Regular
	J3	Cádiz	Very tall	Very good
	J4	Cádiz	Short	Good
	J5	Córdoba	Short	Very good
	J6	Córdoba	Very tall	Bad
	J7	Granada	Short	Bad
	J8	Granada	Very tall	Bad
	J9	Grandad	Tall	Regular
	J10	Huelva	Tall	Very good
	J11	Jaen	Short	Very good
	J12	Jaen	Normal	Regular
	J13	Jaen	Tall	Very good
	J14	Jaen	Tall	Very good
	J15	Málaga	Short	Very good
	J16	Málaga	Tall	Regular
	J17	Málaga	Very tall	Very good
	J18	Sevilla	Short	Good
	J19	Sevilla	Very tall	Good
	J20	Sevilla	Normal	Good

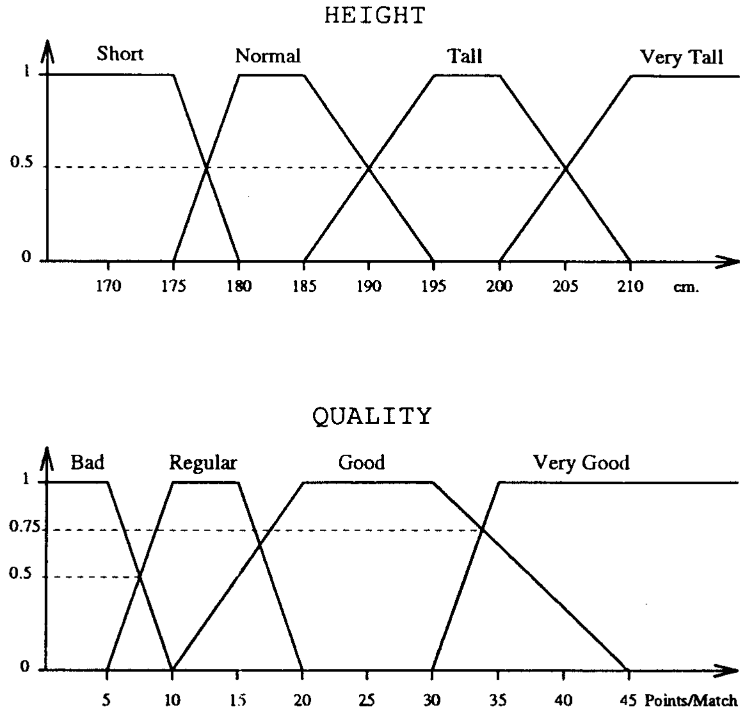

Figure 2. Definition of labels on HEIGHT and QUALITY attributes.

The expression which resolves this query is

$$\{p,t,h \mid \exists q(R(p,t,h,q) \wedge \Theta = (h, \text{Short}) \geq 0.5)\}$$

First, the value component is computed, i.e., the tuples (p,t,h) that satisfy the predicate. Here, nine tuples exist which satisfy the predicate: see the value component of the relation in Table IV. We will denote these nine tuples of the value component as S_r with $r = 1,\ldots,9$. In Table IV, we indicate the substitution name on the right of every tuple. Afterward, to compute the compatibility component it is necessary to study which compatibility attributes exist. We do not consider the attributes C_{PLAYER} and C_{TEAM} in the result since we have: $\forall r = 1,\ldots,9$,

$$c_{r(\text{PLAYER})} = \Delta_p^{S_r}(\psi(p,t,h)) = 1$$

$$c_{r(\text{TEAM})} = \Delta_t^{S_r}(\psi(p,t,h)) = 1$$

Note that, as there is a value for the primary key of R (attribute PLAYER) in S, in order to compute the degree of the WFF it is possible to remove the $\exists c$, substituting the variable c by the only value for each S_i. In other words, as there is a value for the primary key in S, the value of c that obtains the maximum degree [Eq. (24)] will be that which corresponds to the tuple of its primary key.

Table IV. Relation R_3 resulting in Example 3.

$\mathscr{H}$	PLAYER	TEAM	HEIGHT	C_{HEIGHT}	Substitution
$\mathscr{B}$	J2	Almería	Short	1	S_1
	J4	Cádiz	Short	1	S_2
	J5	Córdoba	Short	1	S_3
	J7	Granada	Short	1	S_4
	J11	Jaen	Short	1	S_5
	J12	Jaen	Normal	0.5	S_6
	J15	Málaga	Short	1	S_7
	J18	Sevilla	Short	1	S_8
	J20	Sevilla	Normal	0.5	S_9

The attribute C_{HEIGHT} exists and for each S_r,

$$c_{r(\text{HEIGHT})} = \Sigma_h^{S_r}(\psi(p,t,h))$$

For example, in the last tuple (player "J20") with $r = 9$, we have

$$\begin{aligned} c_{9(\text{HEIGHT})} &= \Delta_h^{S_q}(\psi(p,t,h)) \\ &= \Delta_h^{(\text{J20, Sevilla, Normal})}(\psi(p,t,h)) \end{aligned}$$

Applying the method of Example 2, we obtain

$$\begin{aligned} \Delta^{(\text{J20, Sevilla, Normal})}(\psi(p,t,h)) &\equiv \exists q(R(\text{J20, Sevilla, Normal}, q) \\ &\qquad \wedge \Theta^{=}(\text{Normal, Short})) \\ &\equiv R(\text{J20, Sevilla, Normal, Good}) \wedge 0.5 \\ &\equiv 1 \wedge 0.5 \\ &\equiv 0.5 \end{aligned}$$

We remove the existential quantifier when we replace all the occurrences of the variable q by the Good value, which maximizes the degree of the subformula of $\exists$. In particular, this value of q is the only one which exists in R with the substitution S_9, because this substitution contains a value of the primary key. The degree in this atom, with this value of q, is 1. With other values of q, the degree is 0. As it is a conjunction, we take the smallest degree of both atoms (according to Lemma 2). Thus, with other values of q we have $\min\{0, 0.5\} = 0$ and when q = Good we have $\min\{1, 0.5\} = 0.5$. We take the maximum value [Eq. (24)] of all these values, and we obtain $\max\{0, \ldots, 0, 0.5\} = 0.5$ as the final result. There will be as many zeros as values of c different to Good as there are in $\text{DOM}(\psi)$.

Example 4. Obtain the players with their teams and the heights of those players from *Jaen* or *Málaga* who are *Normal* height (with a minimum degree of 0.5) or *Short* (with a minimum degree of 0.7).

Table V. Relation R_4 resulting from Example 4.

$\mathscr{H}$	PLAYER	TEAM	HEIGHT	C_{HEIGHT}	Substitution
$\mathscr{B}$	J11	Jaen	Short	1	S_1
	J12	Jaen	Normal	1	S_2
	J13	Jaen	Tall	0.5	S_3
	J14	Jaen	Tall	0.5	S_4
	J15	Málaga	Short	1	S_5
	J16	Málaga	Tall	0.5	S_6

The expression to solve this query is as follows,

$$\{p,t,h \mid \exists q\big(R(p,t,h,q) \wedge (t = \text{Jaen} \wedge t = \text{Málaga}) \\ \wedge\big((\Theta^{=}(h,\text{Short}) \geq 0.7) \vee (\Theta^{=}(h,\text{Normal}) \geq 0.5)\big)\big)\}$$

Note that atoms such as $t = \text{Jaen}$ are crisp comparisons and they may appear as fuzzy comparisons of the form $\Theta^{=}(t,\text{Jaen}) \geq 1$. These comparisons take the value 1 if they are true or 0 if they are false. For greater clarity, we write them as crisp comparisons.

The resulting relation, R_4, is shown in Table V. We do not write the attribute C_{TEAM} because its value is 1 for all selected tuples, since for any substitution $S_r = (s_{rp}, s_{rt}, s_{rh})$, we have

$$\begin{aligned}\Delta_t^{S_r}(\psi(p,t,h)) &\equiv \exists q\big(R(s_{rp},s_{rt},s_{rh},q) \wedge (s_{rt} = \text{Jaen} \vee s_{rt} = \text{Málaga}) \\ &\qquad \wedge(\lambda \vee \lambda)\big) \\ &\equiv \max_{q \in \text{DOM}(\psi)} \{1 \wedge (s_{rt} = \text{Jaen} \vee s_{rt} = \text{Málaga}) \wedge \lambda\} \\ &\equiv s_{rt} = \text{Jaen} \vee s_{rt} = \text{Málaga}\end{aligned}$$

In the first step, we remove the $\exists$ by applying Eq. (24). As this predicate is a safe WFF and the only relation used is R, then all the resulting values are in R. Therefore, the degree in the ownership atom will always be equal to 1 for any value of q.

Furthermore, in the resulting relation, the teams (s_{rt}) will be from "Jaen" or from "Málaga," and thus, one of these comparisons will be equal to 1 and the other equal to 0. So, for any substitution S_r, the result will be: $\max\{1,0\} = 1$.

The calculation of C_{HEIGHT} is made in the same way. Let us see how it is calculated for the tuples S_1 and S_6 (Table V),

$$\begin{aligned}c_{1(\text{HEIGHT})} = \Delta_h^{S_1}(\psi(p,t,h)) &\equiv \exists q(R(\text{J11},\text{Jaen},\text{Short},q) \wedge (\lambda \vee \lambda) \\ &\quad \wedge(\Theta^{=}(\text{Short},\text{Short}) \vee \Theta^{=}(\text{Short},\text{Normal}))) \\ &\equiv 1 \wedge \lambda \wedge (1 \vee 0.5) \\ &\equiv \min\{1,\max\{1,0.5\}\} = 1 \\ c_{6(\text{HEIGHT})} = \Delta_h^{s_6}(\psi(p,t,h)) &\equiv \exists q(R(\text{J16},\text{Málaga},\text{Tall},q) \vee (\lambda \vee \lambda) \\ &\quad \wedge(\Theta^{=}(\text{Tall},\text{Short}) \vee \Theta^{=}(\text{Tall},\text{Normal}))) \\ &\equiv 1 \wedge \lambda \wedge (0 \vee 0.5) \\ &\equiv \min\{1,\max\{0,0.5\}\} = 0.5\end{aligned}$$

In the first case the variable q is replaced by "Very good" and in the second case it is replaced by "Regular." These are the only values of q so that the degree of the subformula of $\exists$ is greater than 0.

Example 5. Starting with relations R_3 (Table IV) and R_4 (Table V) of Examples 3 and 4, respectively, the union (expressed in terms of relational algebra) of both relations would correspond to the query: *Get the players with their teams and heights of those players belonging to R_3 or R_4.*

The expression that solves this query is, as in Eq. (11),

$$R_3 \cup R_4 = \{p, t, h \mid R_3(p, t, h) \vee R_4(p, t, h)\}$$

The values of the compatibility attributes C_{PLAYER} and C_{TEAM} will all be equal to 1, and so, we do not show them in the relation in Table VI. For example, we show how these degrees are computed for the tuples S_6, S_{10}, and S_{12}. Note that in order to compute the compatibility degrees, Eq. (20) (or its algorithm) must be applied twice,

$$\begin{aligned}
c_{6(\text{TEAM})} &= \Delta_t^{S_6}(\psi) \equiv \Delta_t^{S_6}(R_3(p,t,h)) \vee \Delta_t^{S_6}(R_4(p,t,h)) \equiv 1 \vee 1 \equiv 1\\
c_{10(\text{TEAM})} &= \Delta_t^{S_{10}}(\psi) \equiv \Delta_t^{S_{10}}(R_3(p,t,h)) \vee \Delta_t^{S_{10}}(R_4(p,t,h)) \equiv 0 \vee 1 \equiv 1\\
c_{12(\text{TEAM})} &= \Delta_t^{S_{12}}(\psi) \equiv \Delta_t^{S_{12}}(R_3(p,t,h)) \vee \Delta_t^{S_{12}}(R_4(p,t,h)) \equiv 1 \vee 1 \equiv 1
\end{aligned}$$

The computation of $c_{r(\text{PLAYER})}$ with $r = 1, \ldots, 12$, is similar to that of $c_{r(\text{TEAM})}$. We now show how to obtain the values of the compatibility attribute C_{HEIGHT} in the same tuples, since we know that this attribute exists in R_3 and in R_4,

$$\begin{aligned}
c_{6(\text{HEIGHT})} &= \Delta_h^{S_6}(\psi) \equiv \min\{0.5, 1\} \vee \min\{1, 1\} \equiv \max\{0.5, 1\} = 1\\
c_{10(\text{HEIGHT})} &= \Delta_h^{S_{10}}(\psi) \equiv 0 \vee \min\{0.5, 1\} \equiv \max\{0, 0.5\} = 0.5\\
c_{12(\text{HEIGHT})} &= \Delta_h^{S_{12}}(\psi) \equiv \min\{0.5, 1\} \vee 0 \equiv \max\{0.5, 0\} = 0.5
\end{aligned}$$

Table VI. Relation $R_5 = R_3 \cup R_4$ of Example 5.

$\mathscr{H}$	PLAYER	TEAM	HEIGHT	C_{HEIGHT}	Substitution
$\mathscr{B}$	J2	Almería	Short	1	S_1
	J4	Cádiz	Short	1	S_2
	J5	Córdoba	Short	1	S_3
	J7	Granada	Short	1	S_4
	J11	Jaen	Short	1	S_5
	J12	Jaen	Normal	1	S_6
	J13	Jaen	Tall	0.5	S_7
	J14	Jaen	Tall	0.5	S_8
	J15	Málaga	Short	1	S_9
	J16	Málaga	Tall	0.5	S_{10}
	J18	Sevilla	Short	1	S_{11}
	J20	Sevilla	Normal	0.5	S_{12}

Example 6. In the same line as in the previous example, the intersection (expressed in terms of relational algebra) of both relations R_3 and R_4 would correspond to the query: *Get all the players with their teams and the heights of those players belonging to R_3 and R_4.*

The expression that solves this query is, as in Eq. (16),

$$R_3 \cap R_4 = \{p, t, h \mid R_3(p, t, h) \wedge R_4(p, t, h)\}$$

The values of the compatibility attributes C_{PLAYER} and C_{TEAM} are always 1, and so they do not appear in the resulting relation in Table VII. The computation of both is very similar: for each substitution S_r with $r = 1, 2, 3$ we have

$$c_{r(\text{TEAM})} = \Delta_t^{S_r}(\psi) \equiv 1 \wedge 1 \equiv 1$$

For the compatibility attribute C_{HEIGHT} we have

$$c_{1(\text{HEIGHT})} = c_{3(\text{HEIGHT})} = \Delta_h^{S_1}(\psi) = \Delta_h^{S_3}(\psi) \equiv \min\{1, 1\} \wedge \min\{1, 1\} \equiv 1$$

$$c_{2(\text{HEIGHT})} = \Delta_h^{S_2}(\psi) \equiv \min\{0.5, 1\} \wedge \min\{1, 1\} \equiv \min\{0.5, 1\} = 0.5$$

Example 7. Show the teams with at least one *Bad* player (with a degree greater than or equal to 0.5),

$$\{t \mid \exists p, h, q(R(p, t, h, q) \wedge \Theta^{=}(q, \text{Bad}) \geq 0.5)\} \tag{29}$$

The result of this query is in Table VIII. For example, the value for C_{TEAM} in the first tuple S_1, $c_{1(\text{TEAM})}$, is obtained by

$$\begin{aligned}\Delta_t^{S_1}(\psi) &\equiv \exists p, h, q\big(\Delta_t^{S_1}(R(p, t, h, q)) \wedge \Delta_t^{S_1}(\Theta^{=}(q, \text{Bad}) \geq 0.5)\big) \\ &\equiv \max\big\{\Delta_t^{S_1}(R(\text{J1}, t, \text{Tall}, \text{Very good}) \wedge \Theta^{=}(\text{Very good}, \text{Bad}) \geq 0.5), \\ &\qquad\qquad \Delta_t^{S_1}(R(\text{J2}, t, \text{Short}, \text{Regular}) \wedge \Theta^{=}(\text{Regular}, \text{Bad}) \geq 0.5)\big\} \\ &\equiv \max\{\min\{1, 0\}, \min\{1, \Theta^{=}(\text{Regular}, \text{Bad})\}\} = \max\{0, 0.5\} = 0.5\end{aligned}$$

Note that when we evaluate the existential quantifier ($\exists$), the bound variables (p, h, and q) are replaced [according to Eq. (24)] by the values of p, h, and q (PLAYER, HEIGHT, and QUALITY) existing in DOM(ψ), in order to take the greatest degree in all these substitutions. In the previous equation we have removed the values of p, h, and q without a tuple in R with TEAM $= S_1$ (Almería), because the degree in the subformula of $\exists$ is 0 for these values.

Table VII. Relation $R_6 = R_3 \cap R_4$ of Example 6.

$\mathscr{H}$	PLAYER	TEAM	HEIGHT	C_{HEIGHT}
$\mathscr{B}$	J11	Jaen	Short	1
	J12	Jaen	Normal	0.5
	J15	Málaga	Short	1

Table VIII. Relation R_7.

$\mathscr{H}$	TEAM	C_{TEAM}
$\mathscr{B}$	Almería	0.5
	Córdoba	1
	Granada	1
	Jaen	0.5
	Málaga	0.5

Calculating $\Delta_t^{S_3}(\psi)$, we find three possible values for the variables p, h, and q with a degree greater than 0 in the subformula of $\exists$. These three values are those in the three R tuples with TEAM = S_3 (Granada),

$$\begin{aligned}
\Delta_t^{S_3}(\psi) &\equiv \exists p,h,q\big(\Delta_t^{S_3}(R(p,t,h,q)) \wedge \Delta_t^{S_3}(\Theta^{=}(q,\text{Bad}) \geq 0.5)\big)\\
&\equiv \max\big\{\Delta_t^{S_3}(R(\text{J7},t,\text{Short},\text{Bad})) \wedge \Delta_t^{S_3}(\Theta^{=}(\text{Bad},\text{Bad}) \geq 0.5),\\
&\qquad \Delta_t^{S_3}(R(\text{J8},t,\text{Very tall},\text{Bad})) \wedge \Delta_t^{S_3}(\Theta^{=}(\text{Bad},\text{Bad}) \geq 0.5),\\
&\qquad \Delta_t^{S_3}(R(\text{J9},t,\text{Tall},\text{Regular})) \wedge \Delta_t^{S_3}(\Theta^{=}(\text{Regular},\text{Bad}) \geq 0.5)\big\}\\
&\equiv \max\{\min\{1,\Theta^{=}(\text{Bad},\text{Bad})\},\min\{1,\Theta^{=}(\text{Bad},\text{Bad})\},\\
&\qquad\qquad \min\{1,\Theta^{=}(\text{Regular},\text{Bad})\}\}\\
&\equiv \max\{\min\{1,1\},\min\{1,1\},\min\{1,0.5\}\} = \max\{1,1,0.5\} = 1
\end{aligned}$$

Applying Eq. (24), we will take the greatest degree of the three and it may be the degree of the tuple with player "J7" or "J8," which both have a degree of 1 (the tuple with the "J9" only has a degree of 0.5).

Example 8. Show teams with at least one *Good* player (with a degree greater than or equal to 0.5),

$$\{t \mid \exists p,h,q(R(p,t,h,q) \wedge \Theta^{=}(q,\text{Good}) \geq 0.5)\} \tag{30}$$

Relation R_8 is in Table IX.

Example 9. Show teams with at least one *Bad* player (with a minimum degree of 0.5) and one *Good* player (with a minimum degree of 0.5),

$$\begin{aligned}\{t \mid \exists p,h,q(R(p,t,h,q) \wedge \Theta^{=}(q,\text{Bad}) \geq 0.5) \wedge \exists p,h,q(R(p,t,h,q)&\\
\wedge\, \Theta^{=}(q,\text{Good}) \geq 0.5)\}&\end{aligned}$$

The result is shown in Table X. Note that the predicate for this example is formed by the conjunction of predicates in Examples 7 and 8. Thus, this query is equivalent to the following,

$$\{t \mid R_7(p,t,h) \wedge R_8(p,t,h)\}$$

Table IX. Relation R_8.

$\mathscr{H}$	TEAM	C_{TEAM}
$\mathscr{B}$	Almería	0.75
	Cádiz	1
	Córdoba	0.75
	Granada	0.66
	Huelva	0.75
	Jaen	0.75
	Málaga	0.75
	Sevilla	1

Example 10. Show the teams in which all its players are *Short* (with a minimum degree of 0.5) or *Good* (with a minimum degree of 0.75),

$$\{t \mid \forall p, h, q(R(p, t, h, q) \rightarrow (\Theta^{=}(h, \text{Short}) \geq 0.5 \vee \Theta^{=}(q, \text{Good}) \geq 0.75))\}$$

The resulting relation is shown in Table XI. Let us show how this result is obtained: If we call the subformula of $\forall\psi_1$, we have

$$\psi(t) = \forall p, h, q\psi_1(p, t, h, q)$$

For example, applying Lemma 4 for the first tuple (Table XI), we will search for the values of (p, h, q) in DOM(ψ) with the least degree in ψ_1 with the substitution S_1. The values of (p, h, q) such that $(p, \text{Almería}, h, q) \in R$ have a degree of 1. So, there are two values for (p, h, q) for which the degree can be less than 1 (the two tuples of the team from Almería). In the following equation we express only these values for (p, h, q),

$$\begin{aligned} c_{1(\text{TEAM})} &= \Delta_t^{S_1}(\psi(t)) \\ &= \min\{\Delta_t^{S_1}(\psi_1(\text{J1}, t, \text{Tall}, \text{Very good})), \Delta_t^{S_1}(\psi_1(\text{J2}, t, \text{Short}, \text{Regular}))\} \end{aligned}$$

Expanding these degrees using Lemma 3 we obtain

$$\begin{aligned} &\Delta_t^{S_1}(\psi_1(\text{J1}, t, \text{Tall}, \text{Very good})) \equiv 1 \rightarrow (0 \vee 0.75) \equiv \max\{1-1, 0.75\} \equiv 0.75 \\ &\Delta_t^{S_1}(\psi_1(\text{J2}, t, \text{Short}, \text{Regular})) \equiv 1 \rightarrow (1 \vee 0.66) \equiv \max\{1-1, 1\} \equiv 1 \end{aligned}$$

Table X. Relation R_9.

$\mathscr{H}$	TEAM	C_{TEAM}
$\mathscr{B}$	Almería	0.5
	Córdoba	0.75
	Granada	0.66
	Jaen	0.5
	Málaga	0.5

Table XI. Relation R_{10}.

$\mathscr{H}$	TEAM	C_{TEAM}
$\mathscr{B}$	Almería	0.75
	Cádiz	0.75
	Huelva	0.75
	Jaen	0.66
	Sevilla	1

and then

$$c_{1(\text{TEAM})} = \Delta_t^{S_1}(\psi(t)) = \min\{0.75, 1\} = 0.75$$

We also obtain the same result by applying Lemma 1: The query is equivalent to

$$\{t \mid \neg \exists p, h, q \neg(\neg R(p, t, h, q) \vee (\Theta^{=}(h, \text{Short}) \geq 0.5 \\ \vee \Theta^{=}(q, \text{Good}) \geq 0.75))\}$$

If we call this WFF without the first negation ψ_2, and this WFF without the quantifier ψ_3, then

$$\psi(t) = \neg \psi_2(t) = \neg \exists p, h, q(\psi_3(p, t, h, q)) = \neg \exists p, h, q(\neg \psi_1(p, t, h, q))$$

Then, in the first tuple

$$c_{1(\text{TEAM})} = \Delta_t^{S_1}(\psi(t)) = 1 - \Delta_t^{S_1}(\psi_2(t))$$

In order to solve $\Delta_t^{S_1}(\psi_2(t))$ we search for values of (p, h, q) in DOM(ψ) with a greater degree in ψ_3 with the substitution S_1. So, we find two values of (p, h, q) for which the degree can be greater than 0. In the following equation we only write these two values of (p, h, q), i.e., the two tuples from Almería,

$$\Delta_t^{S_1}(\eta_2(t)) = \max\{\Delta_t^{S_1}(\psi_3(\text{J1}, t, \text{Tall}, \text{Very good})), \\ \Delta_t^{S_1}(\psi_3(\text{J2}, t, \text{Short}, \text{Regular}))\}$$

Expanding these degrees, we obtain

$$\begin{aligned} \Delta_t^{S_1}(\psi_3(\text{J1}, t, \text{Tall}, \text{Very good})) &\equiv \neg(\neg 1 \vee 0 \vee 0.75) \equiv \neg(0.75) \equiv 0.25 \\ \Delta_t^{S_1}(\psi_3(\text{J2}, \text{Short}, \text{Regular})) &\equiv \neg(\neg 1 \vee 1 \vee 0.66) \equiv \neg(1) \quad\; \equiv 0 \end{aligned}$$

and then

$$\Delta_t^{S_1}(\psi_2(t)) = \max\{0.25, 0\} = 0.25$$

Therefore, we obtain the expected result,

$$c_{1(\mathrm{TEAM})} = \Delta_t^{S_1}(\psi(t)) = 1 - \Delta_t^{S_1}(\psi_2(t)) = 1 - 0.25 = 0.75$$

We also develop the equations for the fourth tuple, S_4,

$$c_{4(\mathrm{TEAM})} = \Delta_t^{S_4}(\psi(t))$$

Now we find four values of (p, h, q) in DOM(ψ) with a degree in ψ_1 with the substitution S_1 smaller than 1 (the four tuples from Jaen). In the following equation we only write these four values of (p, h, q),

$$\begin{aligned}\Delta_t^{S_4}(\psi(e)) = \min\{&\Delta_t^{S_4}(\mathrm{J11}, t, \mathrm{Short}, \mathrm{Very\ good}),\\ &\Delta_t^{S_4}(\psi_1(\mathrm{J12}, t, \mathrm{Normal}, \mathrm{Regular})), \Delta_t^{S_4}(\psi_1(\mathrm{J13}, t, \mathrm{Tall}, \mathrm{Very\ good})),\\ &\Delta_t^{S_4}(\psi_1(\mathrm{J14}, t, \mathrm{Tall}, \mathrm{Very\ good}))\}\end{aligned}$$

Computing all of these degrees, we obtain

$$\begin{array}{llll}\Delta_t^{S_4}(\psi_1(\mathrm{J11}, t, \mathrm{Short}, \mathrm{Very\ good})) & \equiv 1 \to (1 \vee 0.75) & \equiv \max\{1-1, 1\} & \equiv 1\\ \Delta_t^{S_4}(\psi_1(\mathrm{J12}, t, \mathrm{Normal}, \mathrm{Regular})) & \equiv 1 \to (0.5 \vee 0.66) & \equiv \max\{1-1, 0.66\} & \equiv 0.66\\ \Delta_t^{S_4}(\psi_1(\mathrm{J13}, t, \mathrm{Tall}, \mathrm{Very\ good})) & \equiv 1 \to (0 \vee 0.75) & \equiv \max\{1-1, 0.75\} & \equiv 0.75\\ \Delta_t^{S_4}(\psi_1(\mathrm{J14}, t, \mathrm{Tall}, \mathrm{Very\ good})) & \equiv 1 \to (0 \vee 0.75) & \equiv \max\{1-1, 0.75\} & \equiv 0.75\end{array}$$

Thus, we obtain

$$c_{4(\mathrm{TEAM})} = \Delta_t^{S_4}(\psi(t)) = \min\{1, 0.66, 0.75, 0.75\} = 0.66$$

Example 11. Select the teams with players with the same height and quality characteristics as the team from *Cádiz* (with a minimum degree of 0.5).

To solve this query we will create two relations R' and R'' defined by

$$\begin{aligned}R'(t, h, q) &= \{t, h, q \mid \exists p\, R(p, t, h, q)\}\\ R''(h, q) &\equiv \{h, q \mid \exists p, t\ (R(p, t, h, q) \wedge t = \text{Cádiz})\}\end{aligned}$$

where $R(p, t, h, q)$ is the generalized fuzzy relation in Table III.

Expressed in terms of relational algebra, the relation R' is a projection of the relation R onto the attributes TEAM, HEIGHT, and QUALITY. The relation R'' is a selection with the condition TEAM = Cádiz and subsequently a projection onto the attributes HEIGHT and QUALITY. The relations R' and R'' are shown in Tables XII and XIII, respectively.

With relational algebra, this query is solved by the quotient $R' \div R''$, which is equivalent to the following expression in fuzzy calculus, as in Eq. (17),

$$R' \div R'' = \{t \mid \forall h, q\ (R''(h, q) \to R'(t, h, q) \geq 0.5)\}$$

Table XII. Relation R' in Example 11.

$\mathscr{H}$	TEAM	HEIGHT	QUALITY
$\mathscr{B}$	Almería	Tall	Very good
	Almería	Short	Regular
	Cádiz	Very tall	Very good
	Cádiz	Short	Good
	Córdoba	Short	Very good
	Córdoba	Very tall	Bad
	Granada	Short	Bad
	Granada	Very tall	Bad
	Granada	Tall	Regular
	Huelva	Tall	Very good
	Jaen	Short	Very good
	Jaen	Normal	Regular
	Jaen	Tall	Very good
	Málaga	Short	Very good
	Málaga	Tall	Regular
	Málaga	Very tall	Very good
	Sevilla	Short	Good
	Sevilla	Very tall	Good
	Sevilla	Normal	Good

Note that the atom of R' includes a fulfilment threshold ($\gamma = 0.5$). So, tuples partially satisfying the query condition exist in the result. If $\gamma = 1$, only the *teams with players with exactly the same height and quality characteristics as the team from* Cádiz *will be retrieved*.

The result of this query, R_{11}, is shown in Table XIV.

The compatibility degree for the first tuple (S_1) is given by the following equation, we have only shown those values of (h, q) in DOM(ψ) for which the degree of the subformula of $\forall$ is smaller than 1, i.e., the two tuples of R'',

$$
\begin{aligned}
c_{1\text{TEAM}} = \Delta_t^{S_1}(\psi) &= \min\{\Delta_t^{S_1}(R''(\text{Very tall}, \text{Very good}) \\
&\qquad \to R'(t, \text{Very tall}, \text{Very Good})), \\
&\qquad \Delta_t^{S_1}(R''(\text{Short}, \text{Good}) \to R'(t, \text{Short}, \text{Good}))\} \\
&= \min\{\max\{1-1, 0.5\}, \max\{1-1, 0.66\}\} \\
&= \min\{0.5, 0.66\} = 0.5
\end{aligned}
$$

Table XIII. Relation R'' in Examples 11 and 12.

$\mathscr{H}$	HEIGHT	QUALITY
$\mathscr{B}$	Very tall	Very good
	Short	Good

Table XIV. Relation R_{11}.

$\mathscr{H}$	TEAM	C_{TEAM}
$\mathscr{B}$	Almería	0.5
	Cádiz	1
	Jaen	0.5
	Málaga	0.75
	Sevilla	0.75

Briefly, the compatibility degrees in the other tuples are given by

$$\begin{aligned} c_{2\text{TEAM}} &= \min\{\max\{1-1,1\},\max\{1-1,1\}\} &&= 1\\ c_{3\text{TEAM}} &= \min\{\max\{1-1,0.5\},\max\{1-1,0.75\}\} &&= 0.5\\ c_{4\text{TEAM}} &= \min\{\max\{1-1,1\},\max\{1-1,0.75\}\} &&= 0.75\\ c_{5\text{TEAM}} &= \min\{\max\{1-1,0.75\},\max\{1-1,1\}\} &&= 0.75 \end{aligned}$$

One of the advantages of relational calculus versus relational algebra is that any query may be expressed in a single expression in relational calculus. Thus, any algebraic expression, with any number of operators, may be expressed in a single expression in calculus. Consequently, this example could have been solved with a single expression without using the relations R' and R'', and the same result could have been obtained (Table XIV).

$$\{t \mid \forall p', h, q(R(p', \text{Cádiz}, h, q) \rightarrow \exists p R(p, t, h, q) \geq 0.5)\}$$

Fuzzy relational quotient is explained in Ref. 22, obtaining the same result as in relational calculus when $\gamma = 1$. If $\gamma < 1$, in order to obtain identical results using algebra, we must apply a selection after the quotient to retrieve only those tuples surpassing γ.

Example 12. Now, we take a projection (in terms of relational algebra) of R' (Table XII) onto the attributes HEIGHT and QUALITY. The resulting relation, R''', is shown in Table XV and it is obtained by the expression,

$$R'''(h, q) = \{h, q \mid \exists t\, R'(t, h, q)\}$$

Then, starting from relations R''' (Table XV) and R'' (Table XIII), the relational difference, $R''' - R''$ [Eq. (12)], acquires a greater expressivity in relational calculus, solving the following query: *Give me the tuples in R''' and not in R''* (*with a minimum degree of* 0.75),

$$R'''(h, q) - r'' = \{h, q \mid R'''(h, q) \wedge \neg R''(h, q) \geq 0.75\}$$

The result is shown in Table XVI. The computation of the compatibility degrees is easy. For every tuple i, $c_{i(\text{HEIGHT})} = c_{i(\text{QUALITY})}$ is satisfied and, for

Table XV. Relation R''' in Example 12.

$\mathscr{H}$	HEIGHT	QUALITY
$\mathscr{B}$	Tall	Very good
	Short	Regular
	Very tall	Very good
	Short	Good
	Short	Very good
	Very tall	Bad
	Short	Bad
	Tall	Regular
	Normal	Regular
	Very tall	Good
	Normal	Good

example, in the two first tuples (S_1 and S_2) we have

$$\Delta_h^{S_1}(\psi) = \Delta_q^{S_1}(\psi) \equiv 1 \wedge \neg 0.5 \equiv \min\{1, 1-0.5\} = 0.5$$
$$\Delta_h^{S_2}(\psi) = \Delta_q^{S_2}(\psi) \equiv 1 \wedge \neg 0.66 \equiv \min\{1, 1-0.66\} = 0.34$$

Note that for all tuples, the degree in the left part of the conjunction is always 1. Therefore, taking the minimum of the conjunction (Lemma 2) the result will be equal to the degree in the right part of the conjunction. This right part is a negation ($\neg$) and therefore the more a tuple belongs to R'', the less it will belong to the difference and less will be the compatibility degree of its attributes in the result. If a tuple belongs to R'' with a degree greater than 0.75 then it does not belong in the result, because 0.75 is the fulfilment threshold adopted.

VII. CONCLUSIONS

In this paper we have presented a fuzzy relational calculus for fuzzy databases. This calculus is an extension of classic calculus and so it is also useful in classic databases or in relations without fuzzy attributes.

Moreover, we have presented a function Δ that, as we have shown, is an evaluator which returns the degree to which a concrete value x, of a concrete

Table XVI. Relation $R_{12} = R''' - R''$ in Example 12.

$\mathscr{H}$	HEIGHT	C_{HEIGHT}	QUALITY	C_{QUALITY}
$\mathscr{B}$	Tall	0.5	Very good	0.5
	Short	0.34	Regular	0.34
	Very tall	1	Bad	1
	Short	1	Bad	1
	Tall	1	Regular	1
	Normal	0.5	Regular	0.5
	Normal	0.5	Good	0.5

tuple (S, K), satisfies the predicate ψ. This function allows us to ascertain the degree to which every value of every tuple satisfies the query condition, i.e., Δ returns the compatibility attributes in the resulting generalized fuzzy relation.

The work of function Δ is essential, because relational calculus is based on first-order predicate calculus and so, the predicates can only be True or False. This feature will be used, as in the classic model, to determine whether a tuple belongs to the resulting relation: if the values of the tuple make the predicate True, this tuple belongs to the result and if its values make the predicate False, this tuple does not belong to the result. Then, if a tuple satisfies the predicate it is because every value in this tuple satisfies the conditions of the predicate. However, each value has satisfied the conditions to a certain degree in the range $[0, 1]$, and it is essential to know these degrees in fuzzy databases. The work of function Δ is to compute these degrees.

It is easy to translate an expression in fuzzy domain relational calculus to an equivalent expression in a fuzzy tuple relational calculus.

Thus, we achieve the two levels of query languages designed by Codd[2] for relational databases but they are extended to fuzzy relational databases: fuzzy relational algebra (defined by the GEFRED model[15,16,22] and the fuzzy relational calculus which we have shown in this paper.

References

1. Codd, E. F. Commun ACM, 1970, 13, 377–387.
2. Codd, E. F. Relational Completeness of Data Base Sublanguages; Data Base Systems, Courant Computer Science Symposia Series; Prentice-Hall: Englewood Cliffs, NJ, 1972; Vol. 6.
3. Kuhns, J. L. Report RM-5428-PR, 1967; Rand Corp., Santa Mónica, CA.
4. Held, G. D.; Stonebraker, M. R.; Wong, E. NCC 44, 1975.
5. Chamberlin, D. D.; Boyce, R. F. ACM SIGMOD Workshop on Data Description, Access and Control, 1974.
6. Chamberlin et al., D. D. IBM J Res Develop 1976, 20.
7. Lacroix, M.; Pirotte, A. 3rd International Conference on Very Large Data Bases, 1977.
8. Lacroix, M.; Pirotte, A. In Architecture and Models in Data Base Management Systems; Nijssen, G. M. Ed.; North-Holland: Amsterdam, 1977.
9. Pirotte, A.; Wodon, P. RAIRO Inform Comput Sci 1977, 11.
10. Zloof, M. M. NCC, 44, Anaheim, CA, May 1975; AFIPS Press: Montvale, NJ, 1977.
11. Ralph, B. B. Database Management: Developing Application Systems using ORACLE; Prentice-Hall International: Englewood Cliffs, NJ, 1989.
12. Date, C. J. An Introduction to Database Systems; Addison-Wesley: Reading, MA 1986; 4th ed.
13. Cornelio, E. R. Bases de Datos Relacionales; Paraninfo, Madrid, 1988.
14. Ullman, J. D. Principles of Database Systems; Comput. Sci. Press: New York; 1982, 2nd ed.
15. Medina, J. M.; Pons, O.; Vila, M. A. Inf Sci 1994, 76, 87–109.
16. Medina Rodríguez, J. M. Ph.D. Thesis, University of Granada, 1994.
17. Buckles, B. P.; Petry, F. E.; Sachar, H. S. Fuzzy Sets Syst 1989, 29, 327–340.

18. Buckles, B. P.; Petry, F. E. Fuzzy Sets Syst 1982, 7, 213–226.
19. Buckles, B. P.; Petry, F. E. Inf Sci 1984, 34, 145–155.
20. Umano, N. In Fuzzy Information and Decision Processes; Gupta-Sanchez, Ed.; North-Holland: Amsterdam, 1982.
21. Fukami, S.; Umano, M.; Muzimoto, M.; Tanaka, H. IEICE Technical Reports, Vol. 78, No. 233, AL-78-85 (Automata and Language), 1979; pp 65–72.
22. Galindo, J.; Medina, J. M.; Aranda, M. C. VII Congreso Español sobre Techologías y Lógica Fuzzy (ESTYLF'97), 1997; Tarragona, Spain.