Towards the Formalization of Fuzzy Relational Database Queries

Aleksandar Perović

Faculty of Transportation and Traffic Engineering, University of Belgrade, Serbia
pera@sf.bg.ac.yu

Aleksandar Takači

Faculty of Technology, University of Novi Sad, Serbia
atakaci@uns.ns.ac.yu

Srdjan Skrbic

Faculty of Science, University of Novi Sad, Serbia
skhrba@uns.ns.ac.yu

Abstract. The aim of this paper is to give guidelines on how to formalize fuzzy relational database queries using $L\Pi\frac{1}{2}$ *fuzzy logic. After the short introduction, we give an overview of the* $L\Pi\frac{1}{2}$ *logic. In the continuation we give a brief overview of the FRDB queries and query-database similarity relation. We conclude the paper with the description of FRDB query formalization using presented definitions.*

Keywords: FRDB, PFSQL, $L\Pi\frac{1}{2}$ *logic.*

1 Introduction

Fuzzy Relational Databases (FRDB) are introduced in order to overcome the lack of ability of relational databases to model uncertain and incomplete data. The use of fuzzy sets and fuzzy logic to extend existing database models to include these possibilities has been utilized since the 1980s. In [1] and [13], authors offer one of the first approaches to incorporate fuzzy logic in ER model. Their model allows fuzzy attributes in entities and relationships. Furthermore, the FRDB model was developed in [4, 5] i.e. a way to use fuzzy EER model to model the database and represent modeled fuzzy knowledge using relational database in detail was founded. A more complete survey of research in this area can be found in [11]. Following these attempts, in [10, 11, 12] authors defined a new type of fuzzy SQL language based on the FRDB model developed specifically for this purpose.

Formal development of fuzzy logic is a well worked area. Various Hilbert style axiomatizations can be found in [3]. In order to obtain a complete axiomatization of FRDB values, we have used the interpretation method. The aim of this paper is to obtain an interpretation of FRDB queries in an existing fuzzy logic. We found that $L\Pi\frac{1}{2}$ logic provides enough elements to interpret FRDB.

2 $\mathbf{L\Pi\frac{1}{2}}$ Logic

The $L\Pi\frac{1}{2}$ logic is a fuzzy logic that combines the Łukasiewicz logic and the Product logic. The primitive connectives of $L\Pi\frac{1}{2}$ are:

- $\odot$ (product conjunction),
- $\rightarrow_L$ (Łukasiewicz implication),
- $\rightarrow_\Pi$ (product implication),
- truth constants $\overline{0}$ and $\overline{\frac{1}{2}}$.

The axioms and the inference rules of $L\Pi\frac{1}{2}$ can be found in [2, 6, 7].

Semantically, the above connectives are evaluated in the following way:

$$e(\overline{0}) = 0 \ , \ e(\overline{\tfrac{1}{2}}) = \frac{1}{2} \qquad (1)$$

$$e(\varphi \odot \psi) = e(\varphi) \cdot e(\psi) \qquad (2)$$

$$e(\varphi \rightarrow_L \psi) = \min(1, 1 - e(\varphi) + e(\psi)) \qquad (3)$$

$$e(\varphi \rightarrow_{\Pi} \psi) = \begin{cases} 1 & , \quad e(\varphi) \geq e(\psi) \\ \dfrac{e(\psi)}{e(\varphi)} & , \quad e(\varphi) < e(\psi) \end{cases} \tag{4}$$

Note that both Łukasiewicz implication and product implication behave like orderings. The following connectives (we will give them semantically) can be defined in $L\Pi\frac{1}{2}$ (see [7]):

$$e(\neg_L \varphi) = 1 - e(\varphi) \text{ (Łukasiewicz negation)} \tag{5}$$

$$e(\varphi \oplus \psi) = \min(1, e(\varphi) + e(\psi)) \text{ (Łukasiewicz disjunction)} \tag{6}$$

$$e(\varphi \,\&\, \psi) = \max(0, e(\varphi) + e(\psi) - 1) \text{ (Łukasiewicz conjunction)} \tag{7}$$

$$e(\neg_{\Pi} \varphi) = \begin{cases} 1 & , \quad e(\varphi) = 0 \\ 0 & , \quad e(\varphi) > 0 \end{cases} \text{ (product negation)} \tag{8}$$

$$e(\varphi \vee_{\Pi} \psi) = e(\varphi) + e(\psi) - e(\varphi) \cdot e(\psi) \text{ (product disjunction)} \tag{9}$$

$$e(\Delta \varphi) = \begin{cases} 1 & , \quad e(\varphi) = 1 \\ 0 & , \quad e(\varphi) < 1 \end{cases} \tag{10}$$

$$e(\nabla \varphi) = \begin{cases} 1 & , \quad e(\varphi) > 0 \\ 0 & , \quad e(\varphi) = 0 \end{cases} \tag{11}$$

$$e(\varphi \div \psi) = \max(0, e(\varphi) - e(\psi)) \tag{12}$$

$$e(\varphi \wedge \psi) = \min(e(\varphi), e(\psi)) \text{ (Gődel conjunction)} \tag{13}$$

$$e(\varphi \vee \psi) = \max(e(\varphi), e(\psi)) \text{ (Gődel disjunction)} \tag{14}$$

$$e(\varphi \equiv \psi) = 1 - | e(\alpha) - e(\beta) | . \tag{15}$$

Example 1. Each rational number from the real unit interval is definable in $L\Pi\frac{1}{2}$. Indeed, if m, n and k are positive integers such that m<n<k, then m/n can be represented by:

$$\underbrace{(\varphi \oplus \cdots \oplus \varphi)}_{n} \rightarrow_{\Pi} \underbrace{(\varphi \oplus \cdots \oplus \varphi)}_{m} \tag{16}$$

where φ is the formula $\underbrace{\overline{\frac{1}{2}} \odot \cdots \odot \overline{\frac{1}{2}}}_{k}$. Finally, we can represent 1 by $\overline{0} \rightarrow_L \overline{0}$.

3 FRDB Queries

Relational databases (RDB) have been well studied and developed over the years. However, the representation of imprecise, uncertain or inconsistent information is not possible in RDB, thus they require add-ons to handle these types of information. One possible add-on is to allow the attributes to have values that are fuzzy sets on the attribute domain. This direction led to development of fuzzy relational databases (FRDB). From the implementation point of view, values are limited to certain types of fuzzy sets, most often trapezoidal.

In the FRDB model that is being developed at the University of Novi Sad we opted for interval values, triangular and trapezoidal fuzzy numbers and fuzzy quantities. Triangular fuzzy numbers represent imprecise values i.e. "approximately 5". Trapezoidal fuzzy numbers are also called fuzzy intervals. Fuzzy quantities are fuzzy sets with a monotone membership function that have an unbounded kernel from one side. They are used to represent values like "high salary", "short people", "fast cars" etc. In this paper, we allow the attribute values to be trapezoidal fuzzy numbers as it is the case in the most FRDB models.

Fuzzy values of attributes are not incorporated into existing database management systems, meaning that the database management system should be done from scratch. This is a huge task and most often programmers build on existing RDB's. Such is the case with the system developed by the authors also. FRDB usually have their own query language - fuzzy structured query language (FSQL). In [10, 11, 12] we defined our own variant of the FSQL named PFSQL.

The relational model uses a collection of tables to represent data and relationships inside the data. In our model, data values need not be exact. We can handle imprecise and uncertain information using interval values, fuzzy numbers and quantities. For more details see [11]. This information is stored in a fuzzy meta knowledge base, a crucial part of a FRDB. An example of a table from FRDB is given in Figure 1. The value trap(23,27,1,1) represents a trapezoidal fuzzy number whose kernel is [23,27] with a left and right tolerance of 1 and the value tri(1800,100,200) represents a triangular fuzzy number with the center in 1800 and the left and right tolerance of 100 and 200 respectively.

Name	Age	Salary
Istvan	25	trap(1500,1700,100,200)
Dejan	trap(23,27,1,1)	[2000,2100]
Agi	trap(20,22,2,1)	tri(1800,100,200)
Milica	trap(30,35,5,10)	1850

Figure 1

An example of a table from FRDB

Intervals and triangular fuzzy numbers can be viewed as a special case of trapezoidal fuzzy numbers, thus in our interpretation of FRDB using $L\Pi\frac{1}{2}$ logic it is enough to interpret trapezoidal fuzzy numbers. In [8] we have interpret more general fuzzy attribute values via $L\Pi\frac{1}{2}$ logic.

SQL is the most influential commercially marketed database query language. It uses a combination of relational algebra and relational calculus constructs to retrieve desired data from a database. PFSQL is SQL that can handle fuzzy attribute values. The main difference between SQL and PFSQL is that SQL returns a subset of the database as the query result. On the other hand, PFSQL returns a value in the unit interval for each data row. When attributes with fuzzy values appear in the query it is transformed into a query that can be handled by SQL and finally results obtained from the SQL query are then post processed in order to obtain the desired information.

4 Query-DB Similarity Relation

In order to generalize the operator "=" when querying FRDB with FSQL we use fuzzy similarity relations. There are many ways to do this we have opted for the generalization of a well known relation from set theory:

$$(A \subseteq B \wedge B \subseteq A) \Leftrightarrow A = B \tag{17}$$

First, we generalize the subset $\subseteq$ relation.

Definition 10. Let *A*, *B* be two fuzzy sets. The relation *FINCL(A,B)* is defined in the following way:

$$FINCL(A,B) = \frac{card(A \cap B)}{card(A)}. \tag{18}$$

where *card(S)* is the cardinality of the fuzzy set S. Some examples for FINCL relation are given in Table 8.

A	B	$FINCL(A,B)$
tri(170,5,5)	*tri(170,10,10)*	1
tri(170,10,10)	*tri(170,5,5)*	0.375
tri(170,5,5]\)	*tri(175,5,5)*	0.25
tri(170,5,5)	*tri(200,50,50)*	0.635

Figure 2

Examples of FINCL relation

As we have mentioned, the property

$$\text{If } A \subseteq_F B \;\wedge\; B \subseteq_F A \text{ then } A =_F B, \tag{19}$$

holds for relations $\subseteq_F$ and $=_F$. Using this property we will derive the relation *FQ*.

Definition 11. Let *A,B* be two fuzzy sets. The relation $FQ(A,B)$ is defined in the following way:

$$FQ(A,B) = T(FINCL(A,B), FINCL(B,A)) \tag{20}$$

where *T* is a t-norm [9].

If we used $T = T_M$ then we would have:

$$FQ(A,B) = \frac{card(A \cap B)}{\max(card(A), card(B))}. \tag{21}$$

A	*B*	$FQ(A,B)$
tri(25,5,1)	*tri(20,5,5)*	0.25
tri(25,5,25)	*tri(20,5,5)*	0.1
tri(20,1,1)	*tri(21,1,1)*	0.25
tri(20,1,10)	*tri(21,1,1)*	0.011364

Figure 3
Examples of FQ relation

5 FRDB Query Formalization

As we have seen, rational numbers can be represented in $L\Pi\frac{1}{2}$, so hard constraints are expressible in $L\Pi\frac{1}{2}$. In order to capture trapezoidal fuzzy numbers, we will define the following conservative extension (extension by definitions) of $L\Pi\frac{1}{2}$:

For each $0 \le a < b < c < d \le 1$, $a,b,c \in \mathbb{Q}$, we will introduce a new unary connective $[a,b,c,d]$ and add to $L\Pi\frac{1}{2}$ the following axioms:

$$([a,b,c,d]\varphi \equiv \bar{0}) \equiv ((\varphi \rightarrow_L \bar{a}) \vee (\bar{d} \rightarrow_L \varphi)) \tag{22}$$

$$([a,b,c,d]\varphi \equiv \bar{1}) \equiv ((\bar{b} \rightarrow_L \varphi) \wedge (\varphi \rightarrow_L \bar{c})) \tag{23}$$

$$([a,b,c,d]\varphi \equiv ((\varphi \odot \overline{\tfrac{1}{b-a}}) \div \overline{\tfrac{a}{b-a}})) \equiv ((\bar{a} \rightarrow_L \varphi) \wedge (\varphi \rightarrow_L \bar{b})) \tag{24}$$

$$([a,b,c,d]\varphi \equiv (\overline{\tfrac{d}{d-c}} \div (\varphi \odot \overline{\tfrac{1}{d-c}}))) \equiv ((\bar{c} \rightarrow_L \varphi) \wedge (\varphi \rightarrow_L \bar{d}))\ . \tag{25}$$

Notice that (22), (23), (24) and (25) actually formalize the trapezoidal fuzzy number $[a,b,c,d]:[0,1] \rightarrow [0,1]$ defined by:

$$[a,b,c,d](x) = \begin{cases} 0 & , \quad x \leq a \; or \; x \leq c \\ 1 & , \quad b \leq x \leq c \\ \frac{x}{b-a} - \frac{a}{b-a} & , \quad a < x < b \\ \frac{d}{d-c} - \frac{x}{d-c} & , \quad c < x < d \end{cases} . \tag{26}$$

Obviously, in case of $b = c$ we obtain a triangular fuzzy number, similarly if $a = b$ and $c = d$ we obtain an interval and finally if $a = b = c = d$ then we have a crisp set.

For example, let us take the trapezoidal fuzzy number representing height $trap(180,190,10,10)$. If we agree the maximum height is 250 then the connective that interprets this value is $[\frac{180-10}{250}, \frac{180}{250}, \frac{190}{250}, \frac{190+10}{250}]$ i.e. $[0.72,0.76,0.8,0.84]$.

The relation *FQ* can be formalized in the following way. First, it is well known that the cardinality (*card*) of a trapezoidal fuzzy set is actually the area bounded by the membership function of that fuzzy set and the x-axis thus the area of a trapezoidal fuzzy number can calculated in the following way:

$$\text{area([a,b,c,d])} = (c-b) + \frac{(b-a)+(d-c)}{2} \tag{27}$$

Moreover, the intersection of two trapezoidal fuzzy sets need not be a trapezoidal fuzzy number, but the area induced by that set can be calculated using the parameters of the two fuzzy sets. The general algorithm of finding the intersection is very long, but not very complicated. Thus, we can conclude that the formalization of the FQ relation can be done since it can be reduced to the composition of if-then clauses and basic arithmetical operators.

If we reduce the PFSQL to a basic set of operators i.e. the syntax of a basic PFSQL:

```
SELECT attributes
FROM tablenames
WHERE logicalformula
logicalformula:= (trap=trap)|( logicalformula AND
logicalformula)| ( logicalformula OR logicalformula)
```

Since the '=' is interpreted as FQ and the logical connectives "AND", "OR" can be interpreted by $L\Pi\frac{1}{2}$ we conclude that the WHERE line calculation can interpreted via $L\Pi\frac{1}{2}$ logic.

Conclusion

In this paper we have formalized attribute values FRDB using $L\Pi\frac{1}{2}$ and calculation of any logical formula in the WHERE line of PFSQL query. The logic $L\Pi\frac{1}{2}$ has proven to be a tool powerful enough to formalize attribute values in FRDB. In the future research we plan to use our formalization for complexity analysis of FRDB using the results obtained in complexity analysis of different fuzzy logics.

Acknowledgements

Authors would like to acknowledge the support of the Serbian Ministry of Science, project "Mathematical models of non-linearity, uncertainty and decision making", No. 144012 and project "Abstract Methods and Applications in Computer Science" No. 144017A, and also the support of the Provincial Secretariat for Science and Technological Development of Vojvodina.

References

[1] B. Buckles, F. Petry: A Fuzzy Representation of Data for Relational Databases. Fuzzy Sets and Systems, 7 (3), 1982, pp. 213-226

[2] P. Cintula, P. Hájek, R. Horčik: Formal Systems of Fuzzy Logic and Their Fragments. Annals of Pure and Applied Logic, 150, 2007, pp. 40-65

[3] F. Esteva, L. Godo, P. Hájek, M. Navara: Reziduated Fuzzy Logics with an Involutive Negation, Arch. Math. Logic, 39, 2000, pp. 103-124

[4] J. Galindo, J. Medina, M. Aranda: Querying Fuzzy Relational Databases through Fuzzy Domain Calculus. International Journal of Intelligent Systems, 14 (4), 1999, pp. 375-411

[5] J. Galindo, J. Medina, J. Cubero, M. Garcia: Relaxing the Universal

Quantifier of the Division in Fuzzy Relational Databases. International Journal of Intelligent Systems, 16 (6), 2001, pp. 713-742

[6] P. Hájek: Metamathematics of Fuzzy Logic. Kluwer academic publishers, 1998

[7] E. Marchioni, L. Godo: A Logic for Reasoning about Coherent Conditional Probability: A Modal Fuzzy Logic Approach. In J. Leite and J. Alferes, editors, 9th European Conference, Jelia '04, Lecture Notes in Artificial Intelligence (LNCS/LNAI), 3229, 2004, pp. 213-225

[8] A. Perović, A. Takači, S. Škrbić: Interpretation Method in FRDBs. Proceedings of the 6th International Symposium on Intelligent Systems and Informatics (SISY 2008), Subotica, Serbia, Sept 26-27, 2008, pp. 1-3

[9] A. Takači: Schur-Concave Triangular Norms: Characterization and Application in PFCSP. Fuzzy Sets and Systems, 155 (1), 2005, pp. 50-64

[10] A. Takači, S. Škrbić: Priority, Weight and Threshold in Fuzzy SQL Systems. Acta Polytechnica Hungarica, 5(1), 2008, pp. 59-68

[11] A. Takači, S. Škrbić: Data Model of FRDB with Different Data Types and PFSQL. In J. Galindo (Ed.), Handbook of Research on Fuzzy Information Processing in Databases, Hershey, PA: Information Science Reference, 2008, pp. 407-434

[12] S. Škrbić, A. Takači: On Development of Fuzzy Relational Database Applications. Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems (IPMU 2008), Malaga, Spain, 2008, pp. 268-273

[13] A. Zvieli, A., P. Chen: ER Modeling and Fuzzy Databases. Proceedings of the 2nd International Conference on Data Engineering, Los Angeles, CA, 1986, pp. 320-327