University of Ostrava

Institute for Research and Applications of Fuzzy Modeling

The Extraction of Linguistic Knowledge Using Fuzzy Logic and Generalized Quantifiers

Antonín Dvořák and Vilém Novák

Research report No. 44

2002

Submitted/to appear:

Proceedings of FLINS 2002 Conference

Supported by:

Grant IGS 94-1062 of the University of Ostrava

University of Ostrava
Institute for Research and Applications of Fuzzy Modeling
Bráfova 7, 701 03 Ostrava 1, Czech Republic

tel.: +420-69-6160 233 fax: +420-69-6120 478
e-mail: antonin.dvorak@osu.cz

The Extraction of Linguistic Knowledge Using Fuzzy Logic and Generalized Quantifiers

Full version

Antonín Dvořák and Vilém Novák

University of Ostrava,
Institute for Research and Applications of Fuzzy Modeling,
30. dubna 22, 701 03 Ostrava, Czech Republic
E-mail: {antonin.dvorak, vilem.novak}@osu.cz

Abstract

An algorithm for the extraction of linguistic knowledge from data records is presented. Generalized quantifier "many" and weights for individual variables can be included in the statement of the query. General outline of fuzzy logic deduction is introduced. The behavior of the algorithm is illustrated on "difficult order" example.

Keywords: flexible queries; fuzzy logic; generalized quantifiers; IF-THEN rule.

1 Introduction

In this paper we propose an application of fuzzy logic in broader sense[13] (FLb) to the extraction of linguistic knowledge. Our aim is to derive (with the help of an expert) IF-THEN rules describing some complex property of data in a database, for example a "difficult order." The description of such a property may contain also generalized quantifier "many". We use fuzzy logic deduction mechanism based on Łukasiewicz structure of truth values[3, 1, 5]. This inference mechanism is based on the formal theory of fuzzy logic.

Recently a lot of papers has been published on flexible querying in the framework of relational databases[2, 7, 8]. Methods used there are capable of dealing with complex queries such as *"difficult order"* etc. After stating such a query, fuzzy querying system finds items in the database which meet criteria included in the notion "difficult order," such as short delivery time, high freight costs, low order amount etc.

In the paper[4] we proposed an algorithm which extracts linguistic description (a set of IF-THEN rules) describing some complex query (e.g. *difficult order, serious water pollution etc.*) from data.

In our approach we extract linguistic knowledge about orders from an existing database or from incoming data records with the help of an expert who provides several general rules about the query in question. The set of IF-THEN rules obtained in such a way will be called the *linguistic description*. Then, when numerical information about the new order o' appears, it serves us as an input to the fuzzy logical inference mechanism with linguistic description $\mathcal{R}^{\mathrm{gen}}$. The result of it is the fuzzy set which describes the "difficulty" of o'. The most appropriate linguistic description (e.g. *more or less difficult order*) can be then assigned to it by means of *suitable linguistic expression* procedure. However, there can occur the situation that no rule in linguistic description is appropriate for the given input o'. In this situation we use another, "basic" IF-THEN rule $\mathcal{R}^{\mathrm{b}}$ given by the expert for the generation of a new rule $\mathcal{R}_{\mathrm{new}}$, which is then added to the linguistic description, provided that it is not evaluated as superfluous. Linguistic description is in this way continually improved during its use.

We believe that this approach has several advantages with respect to other well-known methods, e.g. neural networks. Namely, we obtain not only an algorithm which can say something about difficulty of orders, but also the linguistic description which summarizes expert knowledge about it and can be used for getting a better insight into what complex queries (such as "difficult order") are.

We propose to extend this algorithm by considering also queries containing quantifiers (*many*, *several*, etc.). The theory of generalized quantifiers is presented in[10], see also[12, 11, 6]. We sketch the theory of generalized quantifiers, apply it to the problem of linguistic knowledge extraction and present some examples and results.

2 Fuzzy logic deduction

Theoretical background of the method used in this paper is *fuzzy logic in broader sense* and, namely logical deduction in the frame of it. This has been described, e.g. in[13], see also[5]. Recall that the formal background is fuzzy logic in narrow sense with evaluated syntax and the set of truth values being Łukasiewicz MV-algebra $\mathcal{L} = \langle [0,1], \vee, \wedge, \otimes, \oplus, 0, 1\rangle$ where $\otimes, \oplus$ are Łukasiewicz operations of conjunction and disjunction giving then rise to the Łukasiewicz implication $a \to b = \neg a \oplus b = 1 \wedge (1 - a + b)$ for all $a, b \in [0,1]$.

We will, furthermore, deal with linguistic description, which is a set

$$\mathcal{R} := \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_r\},$$

where each $\mathcal{R}_i$, $i = 1, 2, \ldots, r$ is an IF-THEN rule

$$\mathcal{R}_i := \text{IF } X_1 \text{ IS } \mathcal{A}_{1i} \text{ AND } \cdots \text{ AND } X_n \text{ IS } \mathcal{A}_{ni} \text{ THEN } Y \text{ IS } \mathcal{B}_i, \tag{1}$$

where $\mathcal{A}_{ji}$, $j = 1, \ldots, n$, $\mathcal{B}_i$ are linguistic expressions taken from some previously defined set $\mathcal{S}$. We will mostly confine ourselves to the so called *simple linguistic evaluating expressions*, which are expressions of the form

$$[\langle\text{sign}\rangle][\langle\text{linguistic modifier}\rangle]\langle\text{atomic term}\rangle. \tag{2}$$

Examples of the simple linguistic evaluating expressions are *more or less medium, significantly small* etc.

The meaning of rules (1) is formalized using the means of the mentioned fuzzy logic in narrow sense. Without going into details, we only mention that the crucial role is there played by the, so called, *evaluated formulas*, which are couples a/A where A is a formula and $a \in [0,1]$ is its syntactic evaluation. The formulas are constructed in a usual way from the atomic ones using connectives and, possibly, quantifiers.

Using evaluated formulas, fuzzy logic in broader sense makes possible to introduce the concepts of *intension, extension* and *possible world.* This is necessary since expressions of natural language can be understood as names of some properties — intensions, which then lead to classes of elements — extensions being realization of the given property in some possible world. Let us note that only the latter are the fuzzy sets usually considered in the applications of fuzzy logic.

For our purpose, we will identify possible world with a structure $\mathcal{D} = \langle D, P_D, \ldots\rangle$ where D is a support (the universe) and P_D are special fuzzy relations assigned to the atomic formulas from the language.

The linguistic modifiers in (2) are of two basic kinds, namely those with narrowing ("extremely, significantly, very") and extending ("more or less, roughly, quite roughly, very roughly") effect. Note that narrowing modifiers make the meaning of the whole expression more precise while widening ones do the opposite. Thus, "very small" is more precise than "small", which, on the other hand, is more precise than "roughly small", etc. A significant role is also played by the sign, which implies the necessity to distinguish between elements being, in some sense, negative from those being positive. We will usually think of negative and positive numbers in the ordinary meaning but in general, this may not always be the case (i.e. "negative" may simply mean elements "smaller than some point").

Linguistic modifiers induce a canonical ordering of the simple evaluating expressions (2) without sign from narrower to wider ones as follows:

$$\begin{array}{l} \text{extremely } \langle\text{atomic term}\rangle \\ \text{significantly } \langle\text{atomic term}\rangle \\ \text{very } \langle\text{atomic term}\rangle \\ \langle\text{atomic term}\rangle \\ \text{more or less } \langle\text{atomic term}\rangle \\ \text{roughly } \langle\text{atomic term}\rangle \\ \text{quite roughly } \langle\text{atomic term}\rangle \\ \text{very roughly } \langle\text{atomic term}\rangle \end{array} \tag{3}$$

The following is the canonical ordering of the signed simple evaluating expressions:

$$
\begin{array}{c}
\text{“negative extremely big”} \\
\text{“negative significantly big”} \\
\vdots \\
\text{“negative extremely small”} \\
\text{“negative roughly zero”} \\
\text{“zero”} \\
\text{“positive roughly zero”} \\
\text{“positive extremely small”} \\
\vdots \\
\text{“positive extremely big”}
\end{array} \tag{4}
$$

We also have to use specific linguistic expression “undefined”, which conforms with everything, i.e. its extension are always all the elements from the given possible world. Its role will become clear in the next section.

The intension of the rules (1) can formally be written as

$$\mathbf{R}_{xyz} := (\mathbf{A}_{x_1,1i} \wedge \cdots \wedge \mathbf{A}_{x_n,ni}) \Rightarrow \mathbf{C}_{y,i}. \tag{5}$$

This means that IF-THEN rules are interpreted as formal implications. Logical deduction then leads to simple formal proof of the form with one of the rules (1)

$$(\mathbf{A}'_{x_1,1i} \wedge \cdots \wedge \mathbf{A}'_{x_n,ni}), (\mathbf{A}_{x_1,1i} \wedge \cdots \wedge \mathbf{A}_{x_n,ni}) \Rightarrow \mathbf{C}_{z,i}, \mathbf{C}'_{z,i} \tag{6}$$

Though this is a concise formal notation, let us only note that both (5) and (6) in fact mean manipulation with (possibly infinite) sets of evaluated formulas (for the details — see [13]). By prime in (6) we denote some modification of the corresponding formulas. In our paper, this will usually be the intension of the special formula *typical example* (of the given linguistic expression), which can formally be written as

$$\{1/A_x[\mathbf{u}_0]\} \tag{7}$$

for some term $\mathbf{u}_0$. Determination of $\mathbf{u}_0$, however, is out of logic.

Let us remark that the machinery outlined above enables us to formalize, besides other, the concept of *linguistic context*. For example, the word “small”, is used in various situations to denote completely different objects (cf., e.g., “small man” and “small mountain”). Yet the general shape of the membership function is in both cases the same since people understand the concept of being “small” equally. This can be explained on the basis of the difference between intension and extension. Therefore, different contexts mean the interpretation of the same intension $\mathbf{A}_x$ in different possible worlds $\mathcal{D}, \mathcal{D}', \ldots$. The result are different extensions, being fuzzy sets in various universes with analogous shape of the membership function. Let us stress that in our case, the possible world is the given database.

Important role plays the operation of finding a *suitable linguistic expression* Suit. This will be considered as an operation defined for each possible world $\mathcal{D}$ and assigning to an element $d \in D$ from its support the most suitable linguistic expression, i.e. an expression for the meaning of which (more precisely, its extension) the given element suits best. Formally, the operation Suit is a function

$$\text{Suit} : D \longrightarrow \mathcal{S}$$

(cf.[5, 1]). There are algorithms solving this problem in the software LFLC used for computation of the results presented in the “difficult order” example.

3 Generalized quantifiers

As the approach to linguistic quantifiers which best fits the problem of linguistic knowledge extraction we regard the standpoint presented by P. Hájek in [10], Chapter 8. (see also[11]). We are mainly interested in the generalized quantifiers intended for use in fuzzy structures (i.e. predicate symbols are interpreted by fuzzy relations). Moreover, we can restrict ourselves to *finite fuzzy models*, where the carrier of the structure is a finite set.

The predicate fuzzy logic in narrow sense mentioned in Section 2 is extended by a symbol $\int$, $\int A(x)\mathrm{d}x$, where A is unary predicate symbol, is to be read as "many x have the property A". The structure for interpretation of fuzzy logic with quantifier many is extended by a special function $\mu : D \longrightarrow [0,1]$ assigning real numbers ("weights") to the elements of D such that $\sum_{d\in D}\mu(d) = 1$.

The truth value of formula $\int B(x)\mathrm{d}x$ in the structure (possible world) $\mathcal{D}$ is computed by

$$\mathcal{D}(\int B(x)\mathrm{d}x) = \sum_{d\in D}\mathcal{D}(B_x[\mathbf{d}])\,\mu(d) \tag{8}$$

where $B_x[\mathbf{d}]$ is a closed instance of a formula B with closed term $\mathbf{d}$, being a name of an element $d \in D$, is substituted for x (for details about syntax and semantics of fuzzy logic in narrow sense see[13], Chapter 4.

4 An algorithm

The database (i.e. the possible world) is supposed to be the set

$$\mathrm{D} = \{(u_{i1}, u_{i2}\ldots, u_{in})|\ i = 1, 2, \ldots, m\},$$

of data records, where $u_{ij}, j = 1, 2, \ldots, n$ are values of attributes. The problem is to extract (with the help of an expert) linguistic description about complex query (*difficult order, serious water pollution, etc.*) and to decide for the individual data record (already present in database, or new one) whether and how much it fits that query. The algorithm presented already in[4] is now modified to allow also the use of quantifier "many", which allows us to work with more realistic queries of type "An order is a difficult one if amount is small, discount is large, delivery time is short, freight cost is big and many of these criteria are fulfilled". Moreover, we can also assign weights to individual criteria to stress or weaken the roles of them.

Our algorithm consists of the following steps:

1. *Providing the basic IF-THEN rule.*
   First, the expert has to specify one IF-THEN rule, which characterize the query and can be taken as a definition of it.

   $$\mathcal{R}^{\mathrm{b}} := \text{IF } X_1 \text{ IS } \mathcal{A}_1^{\mathrm{b}} \text{ AND } \cdots \text{ AND } X_n \text{ IS } \mathcal{A}_n^{\mathrm{b}} \text{ THEN } Y \text{ IS } \mathcal{B}^{\mathrm{b}},$$

   He/she also decides whether the query is quantified. If it is not the case, the algorithm is identical with that presented in[4]. He/she also assigns weights $\mu(i)$, $i = 1, \ldots n$, $\sum_{i=1}^{n}\mu(i) = 1$.

   For "difficult order" example the basic rule takes the form

   $$\begin{aligned}\mathcal{R}^{\mathrm{b}} \;:=\; & \text{IF amount IS } \textit{small} \text{ AND discount IS } \textit{large} \text{ AND} \\ & \text{delivery time IS } \textit{small} \text{ AND freight cost IS } \textit{big} \\ & \text{THEN difficulty IS } \textit{big}.\end{aligned} \tag{9}$$

   It means that difficulty of order is characterized by values of four antecedent variables, namely amount, discount, delivery time and freight cost. Expert's definition of difficult order is, according to (9), that order is difficult if

   - the amount is *small* and

- the discount is *large* and
- the delivery time is *small* and
- the freight cost is *big* and
- many of these criteria should be fulfilled.

If the weights $\mu(i)$ are not given explicitly, we define $\mu(i) = \frac{1}{n}$ for $i = 1, \ldots, n$.

2. *Providing general knowledge.*
Now the expert can provide several IF-THEN rules. These rules express some general conditions which expert asserts to be true. For example, in our "difficult order" example, these can be as follows:

$$\begin{aligned} \mathcal{R}_1 &:= \text{IF amount IS } \textit{small} \text{ THEN difficulty IS } \textit{big}, \\ \mathcal{R}_2 &:= \text{IF delivery time IS } \textit{short} \text{ AND freight cost IS } \textit{big} \\ & \text{THEN difficulty IS } \textit{big}. \\ \mathcal{R}_3 &:= \text{IF delivery time IS } \textit{long} \text{ THEN difficulty IS } \textit{small}. \end{aligned}$$

Note that not all antecedent variables are used in $\mathcal{R}_1$ and $\mathcal{R}_2$. We label the linguistic description with these rules as $\mathcal{R}^{\text{gen}} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_k\}$.

3. *Learning.*
The rules provided by the expert in Step 2 which have included term *undef* (undefined) could be learned more precisely, Also, it can frequently happen in Step 4 (below) that no rule can be applied for the given new or already present database record o′. In these cases we can use rule $\mathcal{R}^{\text{b}}$ for the generation of a new rule, which will be then added to the linguistic description $\mathcal{R}^{\text{gen}}$ and will be therefore applied, when the same (or similar) input record occurs in the future.

   (a) To achieve this, let us consider a new data record $\text{o}' = (u'_1, u'_2, \ldots, u'_n)$. We introduce a new predicate symbol C which will be interpreted by a membership degree of the attribute u'_i in the extension A_i of evaluating linguistic expressions $\mathcal{A}_i^{\text{b}}$ occurring in the antecedent of the rule $\mathcal{R}^{\text{b}}$. Then we form the structure
   $$\mathcal{M}' = \langle M, C_{\mathcal{M}'}, \mu \rangle,$$
   where $M = \{1, 2, \ldots, n\}$, $C_{\mathcal{M}'}(i) = A_i u'_i$, and the function μ was introduced in Step 1.
   Then we compute the truth value of the formula $\int \mathbf{C}(x)\mathrm{d}x$ in the structure $\mathcal{M}'$,
   $$q = \sum_{i=1}^{n} C_{\mathcal{M}'}(i)\mu(i).$$
   Now we find suitable linguistic expression Suit(q) for it and form a new rule
   $$\begin{aligned} \mathcal{R}_{\text{new}} &:= \text{IF } X_1 \text{ IS Suit}(u'_1) \text{ AND } X_2 \text{ IS Suit}(u'_2) \text{ AND} \\ & \ldots \text{ AND } X_n \text{ IS Suit}(u'_n) \text{ THEN } Y \text{ IS Suit}(q) \end{aligned}$$
   Intuitively, when the number q is big, then it indicates that the records o' fits well the query, and vice versa.

   (b) After formation of the rule $\mathcal{R}_{\text{new}}$ we at first check whether it should be inserted into linguistic description $\mathcal{R}^{\text{gen}}$. It will not be inserted, if one of the following happens (cf.[4]):
   - $\mathcal{R}_{\text{new}}$ is identical with a rule in $\mathcal{R}^{\text{gen}}$.
   - Succedent of $\mathcal{R}_{\text{new}}$ is identical with the succedent of some rule $\mathcal{R}_i$ from $\mathcal{R}^{\text{gen}}$ and antecedent of $\mathcal{R}_{\text{new}}$ is narrower than that of $\mathcal{R}_i$.
   - Antecedent of $\mathcal{R}_{\text{new}}$ is identical with the antecedent of some rule $\mathcal{R}_i$ from $\mathcal{R}^{\text{gen}}$ and succedent of $\mathcal{R}_{\text{new}}$ is wider than that of $\mathcal{R}_i$.
   - Succedent of $\mathcal{R}_{\text{new}}$ is *zero*.

4. *Fuzzy logical deduction.*
   After the learning step, we use fuzzy logical deduction with o' as an observation and $\mathcal{R}^{\text{gen}}$ as the linguistic description.

   When the information about a new data record becomes available, i.e. there is a tuple $o' = (u'_1, u'_2 \dots, u'_n)$ at our disposal, we perform one inference step with the extracted linguistic description $\mathcal{R}^{\text{gen}}$. The result is the fuzzy set $B'(o')$ (see Section 2). Then we can construct the most appropriate evaluating expression to it by means of the operation Suit of finding the most suitable linguistic expression
   $$\mathcal{B}' = \text{Suit}(\text{Deff}(B'(o'))),$$
   where $\text{Deff} : \mathcal{F}(X) \to X$ is a defuzzification operation on $\mathcal{F}(X)$ being the set of all fuzzy sets on X.

5 Results

In this section we present linguistic descriptions and other results computed for our "difficult order" example. We keep the same structure as in the previous Section 4.

1. The basic rule $\mathcal{R}^{\text{b}}$ was already presented and explained in the previous section (formula (9)). The weights $\mu(i)$ were set as follows: $\mu(1) = \mu(4) = 0.3$, $\mu(2) = \mu(3) = 0.2$.

2. The general linguistic description $\mathcal{R}^{\text{gen}}$ delivered by the expert is depicted, along with the rule $\mathcal{R}^{\text{b}}$, in Table 1. Note the use of *undef.* The abbreviations assigned to simple linguistic evaluating

Table 1: The general linguistic description $\mathcal{R}^{\text{gen}}$

No.	Amount	Discount	Delivery time	Freight cost	Diff. of order
1	*Sm*	*Bi*	*Sm*	*Bi*	*Bi*
2	*Sm*	*undef*	*undef*	*undef*	*Bi*
3	*undef*	*undef*	*Sm*	*Bi*	*Bi*
4	*undef*	*undef*	*Bi*	*undef*	*Sm*

   expressions can be found in Appendix. The second IF-THEN rule says, roughly speaking, that if the amount of product in question is small, the order is difficult regardless of the other attributes. The third rule expresses the expert's opinion, that if delivery time is short freight costs are big, the order is difficult, again regardless with respect to other attributes. The fourth rule means that if delivery time is long, the order should not be regarded as difficult.

3. Table 2 contains a sample of data records used during the learning process (we suppose all numerical values normalized to the real interval $[0, 1]$, i.e. 0 denotes minimal and 1 maximal value of the variable in question). New rules generated from these data (added to $\mathcal{R}^{\text{gen}}$) are in Table 3. The first rule, for example, can be read as follows:

   > If Amount is *roughly small* and Discount is *big* and Delivery time is *Sm* and Freight costs are *extremely big* then the difficulty of order is *very big.*

   All rules in Table 3 were checked wrt. 3b, Section 4.

4. If the learning step is finished, the linguistic description $\mathcal{R}^{\text{gen}}$ can be used for fuzzy logic deduction (Section 2) whenever a new data record o' appears. Several prototypical data records o' and appropriate results are given in Table 4. The columns "Deff." and "Suit." contain defuzzified values of $B'(o')$ and the most suitable linguistic terms for them, respectively. Note the use of one of rules provided by expert in the third row.

Table 2: Data records used for learning

Amount	Discount	Delivery time	Freight cost
0.20	0.80	0.10	1.00
0.10	0.90	0.20	0.95
0.00	0.95	0.05	0.90
0.20	0.85	0.20	0.95
0.30	1.00	0.10	0.80
0.10	0.85	0.05	1.00
0.05	0.90	0.10	0.95
0.35	0.70	0.20	0.90

Table 3: New rules generated from data in Table 2

No.	Amount	Discount	Delivery time	Freight cost	Diff. of order
5	*RoSm*	*Bi*	*Sm*	*ExBi*	*VeBi*
6	*Sm*	*VeBi*	*RoSm*	*SiBi*	*SiBi*
7	*Ze*	*SiBi*	*SiBi*	*VeBi*	*ExBi*
8	*RoSm*	*Bi*	*RoSm*	*SiBi*	*Bi*
9	*VRSm*	*ExBi*	*Sm*	*Bi*	*MLBi*
10	*Sm*	*Bi*	*SiSm*	*ExBi*	*ExBi*
11	*SiSm*	*VeBi*	*Sm*	*SiBi*	*ExBi*
12	*MLMe*	*RoBi*	*RoSm*	*VeBi*	*QRBi*

Appendix

Names and abbreviations of simple evaluating expressions (see Table 5).

Acknowledgment

The research of the first author was partially supported by the grant IGS 94-1062 of the University of Ostrava.

Table 4: Data records and results of fuzzy logic deduction

Amount	Discount	Delivery time	Freight cost	Deff.	Suit.	Rule
0.142	0.902	0.040	0.922	0.908	*VeBi*	6
0.304	0.900	0.202	0.780	0.481	*VRBi*	12
0.960	0.120	0.840	0.112	0.111	*Sm*	4
0.002	1.000	0.172	0.898	0.988	*ExBi*	11

Table 5: Names and abbreviations of simple evaluating expressions

Abbreviation	Full name	Abbreviation	Full name
Sm	*small*	*ML*	*more or less*
Me	*medium*	*Ro*	*roughly*
Bi	*big*	*QR*	*quite roughly*
Ex	*extremely*	*VR*	*very roughly*
Si	*significantly*	*Ze*	*zero*
Ve	*very*	*undef*	*undefined*

References

[1] Bělohlávek, R., Novák, V. (to appear) Learning Rule Base in Linguistic Expert Systems. Soft Computing.

[2] Bosc, P., Buckles, B. B., Petry, F. E., Pivert, O. (1999) Fuzzy Databases. In: Bezdek, J. C. et al. (Eds.): Fuzzy Sets in Approximate Reasoning and Information Systems. Kluwer, Boston, 403–468.

[3] Dvořák, A. (2000) Preselection of Rules in Fuzzy Logic Deduction. Int J of Uncertainty, Fuzziness and Knowledge-Based Systems 8:563–572.

[4] Dvořák, A., Novák, V. (2000) On the Extraction of Linguistic Knowledge in Databases Using Fuzzy Logic. In: H. L. Larsen et al. (Eds.): Flexible Query Answering Systems. Recent Advances. Physica Verlag, Heidelberg, 445–454.

[5] Dvořák A., Novák V. (submitted) Fuzzy Logic Deduction with Crisp Observations. Soft Computing.

[6] Fodor, J., Yager, R. R. (2000) Fuzzy set-theoretic operations and quantifiers. In: Dubois, D., Prade, H. (Eds.): Fundamentals of Fuzzy Sets. The Handbook of Fuzzy Set Series Vol. 7. Kluwer, Boston.

[7] Kacprzyk J., Zadrożny, S., Ziołkowski, A. (1989) FQUERY III+: a 'human-consistent' database querying system based on fuzzy logic with linguistic quantifiers. Inf Systems 6:443-453.

[8] Kacprzyk, J., Zadrożny, S. (1996) A fuzzy querying interface for a WWW-server-based relational DBMS. In: Proceedings of 6th IPMU Conference Granada, Vol. 1, 19-24

[9] Kacprzyk, J., Zadrożny, S. (1998) Data Mining via Linguistic Summaries of Data: An Interactive Approach. In: Methodologies for Conception, Design and Application of Soft Computing. Proceedings of Iizuka 1998, 668–671.

[10] Hájek, P. (1998) Metamathematics of Fuzzy Logic. Kluwer, Dordrecht.

[11] Hájek, P. (2000) Many. In: Novák, V., Perfilieva, I. (Eds.): Discovering the World with Fuzzy Logic. Physica-Verlag, Heidelberg, 302–309.

[12] Novák, V. (2000) Antonyms and Linguistic Quantifiers in Fuzzy Logic. Fuzzy Sets and Systems 124:335–351.

[13] Novák, V., Perfilieva, I., Močkoř, J. (1999) Mathematical Principles of Fuzzy Logic. Kluwer, Dordrecht.

[14] Petry, F. V. (1996): Fuzzy Databases. Principles and Applications. Kluwer, Boston.