Graded LinClosure*

Radim Bělohlávek and Vilém Vychodil

Department of Computer Science, Palacky University, Olomouc
Tomkova 40, CZ-779 00 Olomouc, Czech Republic
{radim.belohlavek, vilem.vychodil}@upol.cz

Abstract. We present graded extension of the algorithm LinClosure. Graded LinClosure can be used to compute degrees of semantic entailment from sets of fuzzy attribute implications. It can also be used together with graded extension of Ganter's NextClosure algorithm to compute non-redundant bases of data tables with fuzzy attributes. We present foundations, algorithm, preliminary analysis of its complexity, implementation details, and illustrative examples.

1 Introduction

Fuzzy logic is a formal framework for dealing with a particular type of imprecision. A key idea of fuzzy logic is a graded approach to truth in which we allow for truth degrees other than 0 (falsity) and 1 (full truth). This enables us to consider truth of propositions to various degrees, e.g., proposition "Peter is old" can be assigned a degree 0.8, indicating that "Peter is old" is almost (fully) true. One way of looking at the proposition "Peter is old" being true to degree 0.8 is that it expresses a *graded attribute* "being old to degree 0.8" of the *object* "Peter". When dealing with multiple graded attributes, we often need to determine their dependencies. In [2, 4] we have introduced fuzzy attribute implications as particular dependencies between graded attributes in data sets representing objects and their graded attributes (so-called data tables with fuzzy attributes). A fuzzy attribute implication can be seen as a rule of the form "$A \Rightarrow B$", saying "for each object from the data set: if the object has all graded attributes from A, then it has all graded attributes from B". We have proposed several ways to compute, given an input data set represented by a data tables with fuzzy attributes, a minimal set of dependencies describing all dependencies which are valid (true) in the table, see [5] for a survey.

In this paper we focus on computational aspects of one of the algorithms proposed so far. Namely, we show how to compute fixed points of certain fuzzy closure operators that appear in algorithms from [2, 8]. We introduce an extended version of the LinClosure algorithm which is well known from database systems [17]. Compared to the original LinClosure, our extended algorithm,

* Supported by grant No. 1ET101370417 of GA AV ČR, by grant No. 201/05/0079 of the Czech Science Foundation, and by institutional support, research plan MSM 6198959214.

called a GRADED LINCLOSURE (shortly, a GLINCLOSURE) is more versatile (this is discussed in Section 3) while having the same asymptotic time complexity as LINCLOSURE. Since there is a close relationship between dependencies in data tables with fuzzy attributes and data tables over domains with similarity relations, one can also use GLINCLOSURE for determining functional dependencies in data tables over domains with similarity relations which naturally appear in an extension of Codd's relational model which takes into account similarities on domains, see [6, 7].

2 Preliminaries and Motivation

In this section we present preliminaries of fuzzy logic and basic notions of fuzzy attribute implications which will be used in further sections. More details can be found in [1, 12, 14, 16, 18] and [2, 4, 5]. In Section 2.2 we also present motivations for developing LINCLOSURE in fuzzy setting.

2.1 Fuzzy Logic and Fuzzy Set Theory

Since fuzzy logic and fuzzy sets are developed using general scales of truth degrees, we first introduce structures of truth degrees which are used in our approach. Our basic structures of truth degrees will be so-called complete residuated lattices with hedges. A complete residuated lattice with hedge is an algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, {}^*, 0, 1\rangle$ such that $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice with 0 and 1 being the least and greatest element of L, respectively; $\langle L, \otimes, 1\rangle$ is a commutative monoid (i.e. $\otimes$ is commutative, associative, and $a \otimes 1 = 1 \otimes a = a$ for each $a \in L$); $\otimes$ and $\rightarrow$ satisfy so-called adjointness property: $a \otimes b \leq c$ iff $a \leq b \rightarrow c$ $(a, b, c \in L)$; hedge * satisfies, for each $a, b \in L$, (i) $1^* = 1$, (ii) $a^* \leq a$, (iii) $(a \rightarrow b)^* \leq a^* \rightarrow b^*$, and (iv) $a^{**} = a^*$. Each $a \in L$ is called a truth degree. $\otimes$ and $\rightarrow$ are (truth functions of) "fuzzy conjunction" and "fuzzy implication". Hedge * is a (truth function of) logical connective "very true" and properties of hedges have natural interpretations, see [14, 15]. A common choice of $\mathbf{L}$ is a structure with $L = [0, 1]$ (unit interval), $\wedge$ and $\vee$ being minimum and maximum, $\otimes$ being a left-continuous t-norm with the corresponding $\rightarrow$. Three most important pairs of adjoint operations on the unit interval are: Łukasiewicz $(a \otimes b = \max(a + b - 1, 0)$, $a \rightarrow b = \min(1 - a + b, 1))$, Gödel: $(a \otimes b = \min(a, b)$, $a \rightarrow b = 1$ if $a \leq b$, $a \rightarrow b = b$ else), Goguen (product): $(a \otimes b = a \cdot b$, $a \rightarrow b = 1$ if $a \leq b$, $a \rightarrow b = \frac{b}{a}$ else). Complete residuated lattices include also finite structures of truth degrees. For instance, one can put $L = \{a_0 = 0, a_1, \ldots, a_n = 1\} \subseteq [0, 1]$ $(a_0 < \cdots < a_n)$ with $\otimes$ given by $a_k \otimes a_l = a_{\max(k+l-n,0)}$ and the corresponding $\rightarrow$ given by $a_k \rightarrow a_l = a_{\min(n-k+l,n)}$. Such an $\mathbf{L}$ is called a finite Łukasiewicz chain. Another possibility is a finite Gödel chain which consists of L and restrictions of Gödel operations on $[0, 1]$ to L. A special case of a complete residuated lattice with hedge is the two-element Boolean algebra $\langle \{0, 1\}, \wedge, \vee, \otimes, \rightarrow, {}^*, 0, 1\rangle$, denoted by $\mathbf{2}$ (structure of truth degrees of classical logic). Two boundary cases

of hedges are (i) identity, i.e. $a^* = a$ ($a \in L$); (ii) so-called globalization [20]:

$$a^* = \begin{cases} 1 & \text{if } a = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Moreover, for each $\mathbf{L}$ we consider a derived truth function $\ominus$ defined by

$$a \ominus b = a \otimes ((a \to b)^* \to 0). \tag{2}$$

If * is globalization, i.e. if * is defined by (1), we get

$$a \ominus b = \begin{cases} 0 & \text{if } a \leq b, \\ a & \text{else.} \end{cases} \tag{3}$$

Note that the derived truth function $\ominus$ can be seen as a particular subtraction of truth degrees. This is apparent especially in case of globalization, see (3).

Until otherwise mentioned, we assume that $\mathbf{L}$ denotes a complete residuated lattice (with hedge *) which serves as a structure of truth degrees. Using $\mathbf{L}$, we define the following notions. An $\mathbf{L}$-set (a fuzzy set) A in universe U is a mapping $A\colon U \to L$, $A(u)$ being interpreted as "the degree to which u belongs to A". If U is a finite universe $U = \{u_1, \ldots, u_n\}$ then an $\mathbf{L}$-set A in U can be denoted by $A = \{{}^{a_1}\!/\!_{u_1}, \ldots, {}^{a_n}\!/\!_{u_n}\}$ meaning that $A(u_i)$ equals a_i ($i = 1, \ldots, n$). For brevity, we introduce the following convention: we write $\{\ldots, u, \ldots\}$ instead of $\{\ldots, {}^{1}\!/\!_{u}, \ldots\}$, and we also omit elements of U whose membership degree is zero. For example, we write $\{u, {}^{0.5}\!/\!_{v}\}$ instead of $\{{}^{1}\!/\!_{u}, {}^{0.5}\!/\!_{v}, {}^{0}\!/\!_{w}\}$, etc. Let $\mathbf{L}^U$ denote the collection of all $\mathbf{L}$-sets in U. Denote by $|A|$ the cardinality of the support set of A, i.e. $|A| = |\{u \in U \mid A(u) > 0\}|$. The operations with $\mathbf{L}$-sets are defined componentwise. For instance, union of $\mathbf{L}$-sets $A, B \in \mathbf{L}^U$ is an $\mathbf{L}$-set $A \cup B$ in U such that $(A \cup B)(u) = A(u) \vee B(u)$ ($u \in U$); for * being globalization, we get

$$(A \ominus B)(u) = \begin{cases} 0 & \text{if } A(u) \leq B(u), \\ A(u) & \text{else.} \end{cases} \tag{4}$$

For $a \in L$ and $A \in \mathbf{L}^U$, we define an $\mathbf{L}$-set $a \otimes A$ (a-multiple of A) by $(a \otimes A)(u) = a \otimes A(u)$ ($u \in U$). Binary $\mathbf{L}$-relations (binary fuzzy relations) between U and V can be thought of as $\mathbf{L}$-sets in $U \times V$. Given $A, B \in \mathbf{L}^U$, we define $S(A, B) \in L$ by $S(A, B) = \bigwedge_{u \in U} \big(A(u) \to B(u)\big)$. $S(A, B)$ is called a subsethood degree of A in B and it generalizes the classical subsethood relation $\subseteq$ in a fuzzy setting. In particular, we write $A \subseteq B$ iff $S(A, B) = 1$; and $A \subset B$ iff $S(A, B) = 1$ and $A \neq B$. As a consequence, we have $A \subseteq B$ iff $A(u) \leq B(u)$ for each $u \in U$.

2.2 Fuzzy Attribite Implications

Let Y denote a *finite set of attributes.* Each $\mathbf{L}$-set $M \in \mathbf{L}^Y$ of attributes can be seen as a set of graded attributes because M prescribes, for each attribute $y \in Y$, a degree $M(y) \in L$. A *fuzzy attribute implication* (*over attributes* Y) is an expression $A \Rightarrow B$, where $A, B \in \mathbf{L}^Y$ are fuzzy sets of attributes. Fuzzy

attribute implications (FAIs) are the formulas of fuzzy attribute logic. The intuitive meaning we wish to give to $A \Rightarrow B$ is: "if it is (very) true that an object has all (graded) attributes from A, then it has also all (graded) attributes from B". Formally, for an $\mathbf{L}$-set $M \in \mathbf{L}^Y$ of attributes, we define a *truth degree* $||A \Rightarrow B||_M \in L$ *to which* $A \Rightarrow B$ *is true in* M by

$$||A \Rightarrow B||_M = S(A, M)^* \rightarrow S(B, M), \tag{5}$$

where $S(\cdots)$ denote subsethood degrees, see Section 2.1. The degree $||A \Rightarrow B||_M$ can be understood as follows: if M (semantic component) represents presence of attributes of some object, i.e. $M(y)$ is truth degree to which "the object has the attribute $y \in Y$", then $||A \Rightarrow B||_M$ is the truth degree to which "if the object has all attributes from A, then it has all attributes from B", which corresponds to the desired interpretation of $A \Rightarrow B$. Note also that the hedge * present in (5) serves as a modifier of interpretation of $A \Rightarrow B$ and plays an important technical role, see [2, 4, 5] for details. See also [19] for a related approach.

Let T be a set of fuzzy attribute implications. An $\mathbf{L}$-set $M \in \mathbf{L}^Y$ is called a *model of* T if, for each $A \Rightarrow B \in T$, $||A \Rightarrow B||_M = 1$. The set of all models of T will be denoted by $\mathrm{Mod}(T)$, i.e.

$$\mathrm{Mod}(T) = \{M \in \mathbf{L}^Y \mid \text{for each } A \Rightarrow B \in T\colon ||A \Rightarrow B||_M = 1\}. \tag{6}$$

A *degree* $||A \Rightarrow B||_T$ *to which* $A \Rightarrow B$ *semantically follows from* T is defined by

$$||A \Rightarrow B||_T = \textstyle\bigwedge_{M \in \mathrm{Mod}(T)} ||A \Rightarrow B||_M. \tag{7}$$

Described verbally, $||A \Rightarrow B||_T$ is defined to be the degree to which "$A \Rightarrow B$ is true in each model of T". Hence, degrees $||\cdots||_T$ defined by (7) represent degrees of semantic entailment from T. Let us note that degrees $||\cdots||_T$ can also be fully described via the (syntactic) concept of a *provability degree*, see [5] for a survey.

The set $\mathrm{Mod}(T)$ of all models of T form a particular fuzzy closure system in Y, see [8] for details. Thus, for each $\mathbf{L}$-set $M \in \mathbf{L}^Y$ we can consider its closure in $\mathrm{Mod}(T)$ which is then the least model of T containing M. The closure operator associated with $\mathrm{Mod}(T)$ can be described as follows. First, for any set T of FAIs and any $\mathbf{L}$-set $M \in \mathbf{L}^Y$ of attributes define an $\mathbf{L}$-set $M^T \in \mathbf{L}^Y$ of attributes by

$$M^T = M \cup \bigcup\{S(A, M)^* \otimes B \mid A \Rightarrow B \in T\}. \tag{8}$$

Note that if * is globalization, (8) simplifies as follows:

$$M^T = M \cup \bigcup\{B \mid A \Rightarrow B \in T \text{ and } A \subseteq M\}. \tag{9}$$

Using (8), for each $n \in \mathbb{N}_0$ we define a fuzzy set $M^{T_n} \in \mathbf{L}^Y$ of attributes by

$$M^{T_n} = \begin{cases} M & \text{for } n = 0 \\ (M^{T_{n-1}})^T & \text{for } n \geq 1. \end{cases} \tag{10}$$

Finally, we define an operator $cl_T\colon \mathbf{L}^Y \rightarrow \mathbf{L}^Y$ by

$$cl_T(M) = \textstyle\bigcup_{n=0}^{\infty} M^{T_n}. \tag{11}$$

The following assertion shows the importance of cl_T.

Theorem 1 (see [8]). *Let* $\mathbf{L}$ *be a finite residuated lattice with hedge,* T *be a set of fuzzy attribute implications. Then*

(i) cl_T *defined by* (11) *is a fuzzy closure operator;*

(ii) $cl_T(M)$ *is the least model of* T *containing* M*, i.e.* $cl_T(M) \in \mathrm{Mod}(T)$ *and, for each* $N \in \mathrm{Mod}(T)$*, if* $M \subseteq N$ *then* $cl_T(M) \subseteq N$*;*

(iii) $||A \Rightarrow B||_T = S(B, cl_T(A))$. □

Remark 1. Note that Theorem 1 (iii) says that degrees of semantic entailment from sets of fuzzy attribute implications can be expressed as subsethood degrees of consequents of FAIs into least models generated by antecedents of FAIs. Hence, a single model of T suffices to express the degree $||A \Rightarrow B||_T$, cf. definition (7). In other words, an efficient procedure for computing of closures $cl_T(\cdots)$ would give us an efficient procedure to compute degrees of semantic entailment.

Another area in which a closure operator similar to (11) appears is the computation of non-redundant bases of data tables with fuzzy attributes. A *data table with fuzzy attributes* is a triplet $\langle X, Y, I\rangle$ where X is a set of objects, Y is a finite set of attributes (the same as above), and $I \in \mathbf{L}^{X\times Y}$ is a binary $\mathbf{L}$-relation between X and Y assigning to each object $x \in X$ and each attribute $y \in Y$ a degree $I(x, y)$ to which "object x has attribute y". $\langle X, Y, I\rangle$ can be thought of as a table with rows and columns corresponding to objects $x \in X$ and attributes $y \in Y$, respectively, and table entries containing degrees $I(x, y)$. A row of a table $\langle X, Y, I\rangle$ corresponding to an object $x \in X$ can be seen as a set I_x of graded attributes (a fuzzy set of attributes) to which an attribute $y \in Y$ belongs to a degree $I_x(y) = I(x, y)$. Furthermore, a *degree* $||A \Rightarrow B||_{\langle X,Y,I\rangle}$ *to which* $A \Rightarrow B$ *is true in data table* $\langle X, Y, I\rangle$ is defined by

$$||A \Rightarrow B||_{\langle X,Y,I\rangle} = \bigwedge_{x\in X} ||A \Rightarrow B||_{I_x}. \tag{12}$$

By definition, $||A \Rightarrow B||_{\langle X,Y,I\rangle}$ is a degree to which "$A \Rightarrow B$ is true in each row of table $\langle X, Y, I\rangle$", i.e. a degree to which "for each object $x \in X$: if it is (very) true that x has all attributes from A, then x has all attributes from B". T is called *complete in* $\langle X, Y, I\rangle$ if $||A \Rightarrow B||_T = ||A \Rightarrow B||_{\langle X,Y,I\rangle}$, i.e. if, for each $A \Rightarrow B$, a degree to which T entails $A \Rightarrow B$ coincides with a degree to which $A \Rightarrow B$ is true in $\langle X, Y, I\rangle$. If T is complete and no proper subset of T is complete, then T is called a *non-redundant basis of* $\langle X, Y, I\rangle$. Note that both the notions of a complete set and a non-redundant basis refer to a given data table with fuzzy attributes.

In order to describe particular non-redundant bases of data tables with fuzzy attributes we need to recall basic notions of formal concept analysis of data tables with fuzzy attributes [3, 5]. Given a data table $\langle X, Y, I\rangle$, for $A \in \mathbf{L}^X$, $B \in \mathbf{L}^Y$ we define $A^\uparrow \in \mathbf{L}^Y$ and $B^\downarrow \in \mathbf{L}^X$ by $A^\uparrow(y) = \bigwedge_{x\in X}(A(x)^* \to I(x, y))$ and $B^\downarrow(x) = \bigwedge_{y\in Y}(B(y) \to I(x, y))$, respectively. Operators $^\downarrow, ^\uparrow$ form so-called Galois connection with hedge, see [3]. The set of all fixed points of $^\downarrow, ^\uparrow$ (so-called fuzzy concepts) hierarchically ordered by the subconcept-superconcept relation is called a *fuzzy concept lattice with hedge*, see [3, 5]. A crucial role in determining

a non-redundant basis of a given $\langle X, Y, I \rangle$ is played by an operator which is a modification of cl_T, see (11). The modified operator can be described as follows. For $M \in \mathbf{L}^Y$ put

$$M^{T^*} = M \cup \bigcup \{S(A, M)^* \otimes B \mid A \Rightarrow B \in T \text{ and } A \neq M\}. \tag{13}$$

If * is globalization, (13) is equivalent to

$$M^{T^*} = M \cup \bigcup \{B \mid A \Rightarrow B \in T \text{ and } A \subset M\}. \tag{14}$$

We can now define an operator cl_{T^*} in much the same way as cl_T:

$$M^{T^*_n} = \begin{cases} M & \text{for } n = 0 \\ (M^{T^*_{n-1}})^{T^*} & \text{for } n \geq 1, \end{cases} \tag{15}$$

$$cl_{T^*}(M) = \bigcup_{n=0}^{\infty} M^{T^*_n}. \tag{16}$$

For cl_{T^*} defined by (15), we have the following

Theorem 2 (see [2, 4, 5]). *Let* $\mathbf{L}$ *be a finite residuated lattice with globalization,* $\langle X, Y, I \rangle$ *be a data table with fuzzy attributes. Then there is* T *such that*

(i) cl_{T^*} *is a fuzzy closure operator;*

(ii) *a set of FAIs defined by* $\{P \Rightarrow P^{\downarrow\uparrow} \mid P = cl_{T^*}(P) \text{ and } P \neq P^{\downarrow\uparrow}\}$ *is a non-redundant basis of* $\langle X, Y, I \rangle$. □

Remark 2. From Theorem 2 it follows that for * being globalization a non-redundant basis of $\langle X, Y, I \rangle$ is determined by particular fixed points of cl_{T^*}, namely, by fuzzy sets $P \in \mathbf{L}^Y$ of attributes such that $P = cl_{T^*}(P)$ and $P \neq P^{\downarrow\uparrow}$. Notice that we have not specified the set T of fuzzy attribute implications which is used by cl_{T^*}. A detailed description of that set is outside the scope of our paper, see [2, 4, 5]. Let us just mention that T is computationally tractable. Fuzzy sets of attributes satisfying $P = cl_{T^*}(P)$ and $P \neq P^{\downarrow\uparrow}$ will be occasionally referred to as *pseudo-intents* or *pseudo-closed fuzzy set of attributes.* This is for the sake of consistency with [2, 4, 5], cf. also [9–11, 13].

3 Graded LinClosure

Throughout this section, we assume that $\mathbf{L}$ is a finite linearly ordered residuated lattice with globalization, see (1). Structure $\mathbf{L}$ represents a finite linear scale of truth degrees.

Problem Setting Given a fuzzy set $M \in \mathbf{L}^Y$ of attributes we wish to compute its closures $cl_T(M)$ and $cl_{T^*}(M)$ defined by (11) and (16), respectively. First, note that cl_T and cl_{T^*} differ only in non-strict/strict fuzzy set inclusions "$\subseteq$" and "$\subset$" used in (9) and (14). A direct method to compute $cl_T(M)$ and $cl_{T^*}(M)$, which is given by the definitions of cl_T and cl_{T^*}, leads to an algorithm similar to Closure which is known from database systems [17]. In more detail: for a given M, we iterate through all FAIs in T and for each $A \Rightarrow B \in T$ we test if $A \subseteq M$

($A \subset M$); if so, we add B to M (i.e., we set $M := M \cup B$) and repeat the process until M cannot be enlarged; the resulting M is the closure under cl_T (cl_{T^*}) of the original fuzzy set M. Clearly, this procedure is sound. Let n be the number of attributes in Y and p be the number of FAIs in T. In the worst case, we have to make p^2 iterations in order to compute the closure because in each loop through all FAIs in T there can be only one $A \Rightarrow B$ such that $A \subseteq M$ ($A \subset M$) and $B \not\subseteq M$ (i.e., only one FAI from T causes M to be enlarged). Moreover, for each $A \Rightarrow B$ we need n steps to check the non-strict/strict subsethood $A \subseteq M$ ($A \subset M$). To sum up, the complexity of this algorithm is $O(np^2)$, where n is the number of attributes and p is the number of FAIs from T (cf. [17]).

In this section we present an improved version of the algorithm, so-called GLinClosure (Graded LinClosure), which computes $cl_T(M)$ and $cl_{T^*}(M)$ with complexity $O(n)$, where n is the size of the input. GLinClosure uses each FAI from T only once and allows us to check the non-strict/strict inclusions $A \subseteq M$ ($A \subset M$) which appear in (9) and (14) in a constant time. Our algorithm results by extending LinClosure [17] so that (i) we can use fuzzy sets of attributes instead of classical sets (this brings new technical problems with efficient comparing of truth degrees and checking of strict inclusion, see below), (ii) we can use the algorithm to compute systems of pseudo-intents (i.e., fixed points of cl_{T^*}), and thus non-redundant bases (the original LinClosure [17] cannot be used to compute pseudo-intents [11], it can only compute fixed points of cl_T), this also brings technical complications since we have to maintain a "waitlist of attributes" which can possibly be updated (or not) in future iterations. In what follows we present a detailed description of the algorithm.

Input and Output of the Algorithm The input for GLinClosure consists of a set T of fuzzy attribute implications over Y, a fuzzy set $M \in \mathbf{L}^Y$ of attributes, and a flag $PCLOSED \in \{false, true\}$. The meaning of $PCLOSED$ is the following. If $PCLOSED$ is set to *true*, the output of GLinClosure is $cl_{T^*}(M)$ (the least fixed point of cl_{T^*} which contains M); if $PCLOSED$ is set to *false*, the output of GLinClosure is $cl_T(M)$ (the least fixed point of cl_T, i.e. the least model of T, which contains M).

Representation of Graded Attributes During the computation, we represent fuzzy sets ($\mathbf{L}$-sets) of attributes in Y by ordinary sets of tuples $\langle y, a\rangle$, where $y \in Y$ and $a \in L$. Namely, a fuzzy set $\{{}^{a_1}/_{y_1}, {}^{a_2}/_{y_2}, \ldots, {}^{a_n}/_{y_n}\}$ ($a_1 \neq 0, \ldots, a_n \neq 0$) will be represented by an ordinary set $\{\langle y_1, a_1\rangle, \langle y_2, a_2\rangle, \ldots, \langle y_n, a_n\rangle\}$. We will use both notations $A(y) = a$ and $\langle y, a\rangle \in A$. Whenever we consider $\langle y, a\rangle$, we assume $a \neq 0$. From the implementational point of view, we may represent fuzzy sets of attributes by lists of tuples $\langle y, a\rangle$ instead of sets of such tuples. In such a case, we write $(\langle y_1, a_1\rangle, \langle y_2, a_2\rangle, \ldots, \langle y_n, a_n\rangle)$ instead of $\{\langle y_1, a_1\rangle, \langle y_2, a_2\rangle, \ldots, \langle y_n, a_n\rangle\}$.

Quick Test of Subsethood We avoid repeated testing of inclusions in (9) and (14) analogously as in the original LinClosure. For each fuzzy attribute implication $A \Rightarrow B$ from T we keep a record of the number of attributes due to which A is not contained in the constructed closure. If this number reaches zero, we get that $A \subseteq M$ and we can process $A \Rightarrow B$, see (9). This suffices to check non-

strict subsethood which is needed to compute fixed points of cl_T. In order to check strict subsethood which is needed to compute $cl_{T^*}(M)$, we need to have a quick test to decide if $A \subset M$ provided we already know that $A \subseteq M$. The test can be done with the following notion of cardinality of fuzzy sets. Take a fixed monotone injective mapping $f_{\mathbf{L}} : L \to [0, 1]$. That is, $f_{\mathbf{L}}$ is injective, and for each $a, b \in L$, if $a \leq b$ then $f_{\mathbf{L}}(a) \leq f_{\mathbf{L}}(b)$. For each fuzzy set $M \in \mathbf{L}^Y$ of attributes define a number $\mathrm{card}(M) \in [0, \infty)$ by

$$\mathrm{card}(M) = \textstyle\sum_{\langle y,a\rangle \in M} f_{\mathbf{L}}(a). \tag{17}$$

For instance, if L is a subset of $[0, 1]$, we can put $f_{\mathbf{L}}(a) = a$ $(a \in L)$, and thus $\mathrm{card}(M) = \sum_{\langle y,a\rangle \in M} a$. It is easily seen that if $A \subseteq M$, then $A \subset M$ iff $\mathrm{card}(A) < \mathrm{card}(M)$.

Data Structures Used During the Computation:

NEWDEP is a fuzzy set of attributes which is the closure being constructed;
CARDND is the cardinality of *NEWDEP*;
COUNT$[A \Rightarrow B]$ is a nonnegative integer indicating the number of attributes from A such that $A(y) > NEWDEP(y)$, *COUNT*$[A \Rightarrow B] = 0$ means that A is a subset of *NEWDEP*;
CARD$[A \Rightarrow B]$ is a number indicating cardinality of A, it is used to decide if A is strictly contained in *NEWDEP* when *COUNT*$[A \Rightarrow B]$ reaches zero;
UPDATE is a fuzzy set of attributes which are waiting for update;
WAITLIST is a list of (pointers to) fuzzy sets of attributes which can be added to *NEWDEP* as soon as *NEWDEP* will increase its cardinality; this is necessary if *PCLOSED* = *true*, *WAITLIST* is not used if *PCLOSED* = *false*;
LIST$[y]$ is an attribute-indexed collection of (pointers to) FAIs from T;
SKIP$[y][A \Rightarrow B] \in \{false, true\}$ indicates whether attribute y has already been updated in $A \Rightarrow B$ (*SKIP*$[y][A \Rightarrow B] = true$) or not (*SKIP*$[y][A \Rightarrow B] = false$); *SKIP* is necessary in graded setting to avoid updating an attribute twice which may result in incorrect values of *COUNT*$[A \Rightarrow B]$;
DEGREE$[y][A \Rightarrow B]$ represents a degree in which attribute y is contained in A;
`()` denotes the empty list.

Remark 3. Note that *NEWDEP*, *UPDATE*, *ADD*, *COUNT*, and *LIST* play a similar role as in LinClosure, cf. [17].

Description of the Algorithm and Complexity Analysis The algorithm is depicted in Fig. 1. Due to the limited scope of the paper, we postpone the proof of correctness to a full version of the paper. Let us mention that the algorithm has two basic parts: (i) *initialization of data structures* (lines 1–14), and (ii) *main computation* (lines 15–36). Denote by n the length of the input, and denote by k the number of truth degrees, i.e. $k = |L|$. The initialization can be done with time complexity $O(n)$ or $O(kn)$ depending on data structures we use to store information about *LIST*, *SKIP*, *DEGREE*, *COUNT*, and *CARD*. In either case, we have in fact $O(n)$ because k is a multiplicative constant (size of $\mathbf{L}$ is fixed

Input: a set T of FAIs over Y, a fuzzy set $M \in \mathbf{L}^Y$ of attributes,
and a flag $PCLOSED \in \{false, true\}$
Output: $cl_{T^*}(M)$ if $PCLOSED = true$, or $cl_T(M)$ if $PCLOSED = false$

Initialization:

```
1  NEWDEP := M
2  for each A ⇒ B ∈ T:
3    if A = ∅:
4      NEWDEP := NEWDEP ∪ B
5    else:
6      COUNT[A ⇒ B] := |A|
7      CARD[A ⇒ B] := card(A)
8      for each ⟨y, a⟩ ∈ A:
9        add A ⇒ B to LIST[y]
10       DEGREE[y][A ⇒ B] := a
11       SKIP[y][A ⇒ B] := false
12 UPDATE := NEWDEP
13 CARDND := card(NEWDEP)
14 WAITLIST := ()
```

Computation:

```
15 while UPDATE ≠ ∅:
16   choose ⟨y, a⟩ ∈ UPDATE
17   UPDATE := UPDATE − {⟨y, a⟩}
18   for each A ⇒ B ∈ LIST[y] such that
            SKIP[y][A ⇒ B] = false and DEGREE[y][A ⇒ B] ≤ a:
19     SKIP[y][A ⇒ B] = true
20     COUNT[A ⇒ B] := COUNT[A ⇒ B] − 1
21     if COUNT[A ⇒ B] = 0 and
            (PCLOSED = false or CARD[A ⇒ B] < CARDND):
22       ADD := B ⊖ NEWDEP
23       CARDND := CARDND + Σ_{⟨y,a⟩∈ADD} f_L(a) − f_L(NEWDEP(y))
24       NEWDEP := NEWDEP ∪ ADD
25       UPDATE := UPDATE ∪ ADD
26       if PCLOSED = true and ADD ≠ ∅:
27         while WAITLIST ≠ ():
28           choose B ∈ WAITLIST
29           remove B from WAITLIST
30           ADD := B ⊖ NEWDEP
31           CARDND := CARDND + Σ_{⟨y,a⟩∈ADD} f_L(a) − f_L(NEWDEP(y))
32           NEWDEP := NEWDEP ∪ ADD
33           UPDATE := UPDATE ∪ ADD
34     if COUNT[A ⇒ B] = 0 and PCLOSED = true and
            CARD[A ⇒ B] = CARDND:
35       add B to WAITLIST
36 return NEWDEP
```

Fig. 1. Graded LinClosure

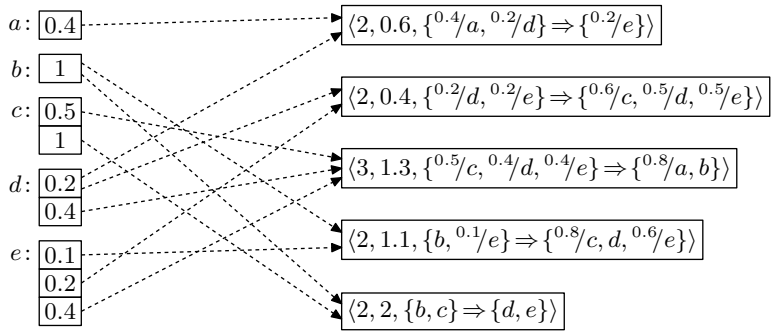

Fig. 2. T-structure encompassing *LIST*, *SKIP*, *DEGREE*, *COUNT*, and *CARD*

and does not depend on the length of the input). In the second part (computation), each graded attribute $\langle y, a\rangle$ is considered at most once for update. Thus, using analogous arguments as in case of the original LinClosure [17], we get that GLinClosure works with asymptotic time complexity $O(n)$. Needless to say, the time complexity of an implementation of GLinClosure depends of our choice of data structures, see Section 4 for further comments.

Remark 4. If $\mathbf{L}$ (our structure of truth degrees) is a two-element Boolean algebra, i.e. if $L = \{0, 1\}$, GLinClosure with *PCLOSED* set to *false* produces the same results as LinClosure [17] (the only difference is that our algorithm allows also for FAIs of the form $\{\} \Rightarrow B$ whereas the original LinClosure does not). From this point of view, GLinClosure is a generalization of LinClosure. GLinClosure is more versatile (even in crisp case): GLinClosure can be used to compute pseudo-intents (and thus a non-redundant basis of data tables with fuzzy attributes) which cannot be done with the original LinClosure (without additional modifications).

4 Implementation Details, Examples, and Remarks

As mentioned before, the efficiency of an implementation of GLinClosure is closely connected with data structures. The information contained in *LIST*, *SKIP*, *DEGREE*, *COUNT*, and *CARD* can be stored in a single efficient data structure. This structure, called a *T-structure*, is a particular attribute-indexed vector of lists of pointers to structures carrying values from *COUNT* and *CARD*. We illustrate the construction of a T-structure by an example. Consider a set T of FAIs which consists of the following fuzzy attribute implications:

φ_1: $\{\} \Rightarrow \{{}^{0.4}/a, {}^{0.1}/d\}$,
φ_2: $\{{}^{0.4}/a, {}^{0.2}/d\} \Rightarrow \{{}^{0.2}/e\}$,
φ_3: $\{{}^{0.2}/d, {}^{0.2}/e\} \Rightarrow \{{}^{0.6}/c, {}^{0.5}/d, {}^{0.5}/e\}$,
φ_4: $\{{}^{0.5}/c, {}^{0.4}/d, {}^{0.4}/e\} \Rightarrow \{{}^{0.8}/a, b\}$,
φ_5: $\{b, {}^{0.1}/e\} \Rightarrow \{{}^{0.8}/c, d, {}^{0.6}/e\}$,
φ_6: $\{b, c\} \Rightarrow \{d, e\}$.

Since φ_1 is of the form $\{\} \Rightarrow B$, its right-hand side is added to *NEWDEP* and the implication itself is not contained in *LIST* and other structures. The

other formulas, i.e. $\varphi_2, \ldots, \varphi_6$, are used to build a new T-structure which is depicted in Fig. 2. The T-structure can be seen as consisting of two main parts. First, a set of records encompassing information about the FAIs, $COUNT$, and $CARD$. For each FAI φ_i, we have a single record, called a T-record, of the form $\langle COUNT[\varphi_i], CARD[\varphi_i], \varphi_i \rangle$, see Fig. 2 (right). Hence, this part of the T-structure also carries information from $LIST$. Second, an attribute-indexed vector of lists containing truth degrees and pointers to T-records, see Fig. 2 (left). A list which is indexed by attribute $y \in Y$ will be called a y-list. The aim of this part of the structure is to keep information about the occurrence of graded attributes that appear in left-hand sides of FAIs from T. In more detail, a y-list contains truth degree $a \in L$ iff there is at lest one $A \Rightarrow B \in T$ such that $0 \neq A(y) = a$. Moreover, if a y-list contains a as its element, then it is connected via pointer to all T-records $\langle m, n, C \Rightarrow D \rangle$ such that $C(y) = a$. Because of the computational efficiency, each y-list is sorted by truth degrees in the ascendant manner. Note that pointers between elements of lists Fig. 2 (left) and T-records Fig. 2 (right) represent information in $SKIP$ ($SKIP[y][A \Rightarrow B] = \mathit{false}$ means that pointer from element $A(y)$ of y-list to T-record of $A \Rightarrow B$ is present). As one can see, a T-structure can be constructed by a sequential updating of the structure with time complexity $O(kn)$. In the following examples, we will use a convenient notation for writing T-structures which correspond in an obvious way with graphs of the from of Fig. 2. For example, instead of Fig. 2, we write:

a: $[(0.4, \langle 2, 0.6, \varphi_2 \rangle)]$
b: $[(1, \langle 2, 2, \varphi_6 \rangle, \langle 2, 1.1, \varphi_5 \rangle)]$
c: $[(0.5, \langle 3, 1.3, \varphi_4 \rangle), (1, \langle 2, 2, \varphi_6 \rangle)]$
d: $[(0.2, \langle 2, 0.4, \varphi_3 \rangle, \langle 2, 0.6, \varphi_2 \rangle), (0.4, \langle 3, 1.3, \varphi_4 \rangle)]$
e: $[(0.1, \langle 2, 1.1, \varphi_5 \rangle), (0.2, \langle 2, 0.4, \varphi_3 \rangle), (0.4, \langle 3, 1.3, \varphi_4 \rangle)]$

Example 1. Consider T which consists of $\varphi_1, \ldots, \varphi_6$ as above in this section. Let $M = \{{}^{0.2}\!/\!d\}$, and $PCLOSED = \mathit{false}$. After the initialization (line 14 of the algorithm), we have $NEWDEP = \{{}^{0.4}\!/\!a, {}^{0.2}\!/\!d\}$ and $UPDATE = (\langle a, 0.4 \rangle, \langle d, 0.2 \rangle)$. Recall that during the update, values of $COUNT$ and $SKIP$ are changed. Namely, values of $COUNT$ may be decremented and values of $SKIP$ are changed to *true*. The latter update is represented by removing pointers from the T-structure. After the update of $\langle a, 0.4 \rangle$ and $\langle d, 0.2 \rangle$, the T-record $\langle 0, 0.6, \varphi_2 = \{{}^{0.4}\!/\!a, {}^{0.2}\!/\!d\} \Rightarrow \{{}^{0.2}\!/\!e\} \rangle$ of φ_2 is processed because we have $COUNT[\varphi_2] = 0$ (see the first item of the T-record). At this point, the algorithm is in the following state:

b: $[(1, \langle 2, 2, \varphi_6 \rangle, \langle 2, 1.1, \varphi_5 \rangle)]$ $\quad ADD = (\langle e, 0.2 \rangle)$
c: $[(0.5, \langle 3, 1.3, \varphi_4 \rangle), (1, \langle 2, 2, \varphi_6 \rangle)]$ $\quad NEWDEP = \{{}^{0.4}\!/\!a, {}^{0.2}\!/\!d, {}^{0.2}\!/\!e\}$
d: $[(0.4, \langle 3, 1.3, \varphi_4 \rangle)]$ $\quad UPDATE = (\langle e, 0.2 \rangle)$
e: $[(0.1, \langle 2, 1.1, \varphi_5 \rangle), (0.2, \langle 1, 0.4, \varphi_3 \rangle), (0.4, \langle 3, 1.3, \varphi_4 \rangle)]$

As a further step of the computation, an update of $\langle e, 0.2 \rangle$ is performed and then the T-record $\langle 0, 0.4, \varphi_3 = \{{}^{0.2}\!/\!d, {}^{0.2}\!/\!e\} \Rightarrow \{{}^{0.6}\!/\!c, {}^{0.5}\!/\!d, {}^{0.5}\!/\!e\} \rangle$ of φ_3 is processed:

b: $[(1, \langle 2, 2, \varphi_6 \rangle, \langle 1, 1.1, \varphi_5 \rangle)]$ $\quad ADD = (\langle c, 0.6 \rangle, \langle d, 0.5 \rangle, \langle e, 0.5 \rangle)$
c: $[(0.5, \langle 3, 1.3, \varphi_4 \rangle), (1, \langle 2, 2, \varphi_6 \rangle)]$ $\quad NEWDEP = \{{}^{0.4}\!/\!a, {}^{0.6}\!/\!c, {}^{0.5}\!/\!d, {}^{0.5}\!/\!e\}$
d: $[(0.4, \langle 3, 1.3, \varphi_4 \rangle)]$ $\quad UPDATE = (\langle c, 0.6 \rangle, \langle d, 0.5 \rangle, \langle e, 0.5 \rangle)$
e: $[(0.4, \langle 3, 1.3, \varphi_4 \rangle)]$

Right after the update of $\langle c, 0.6\rangle$, $\langle d, 0.5\rangle$, and $\langle e, 0.5\rangle$, the algorithm will process the T-record of φ_4. After that, we have the following situation:

b: $[(1, \langle 2, 2, \varphi_6\rangle, \langle 1, 1.1, \varphi_5\rangle)]$
c: $[(1, \langle 2, 2, \varphi_6\rangle)]$

$ADD = (\langle a, 0.8\rangle, \langle b, 1\rangle)$
$NEWDEP = \{{}^{0.8}/a, b, {}^{0.6}/c, {}^{0.5}/d, {}^{0.5}/e\}$
$UPDATE = (\langle a, 0.8\rangle, \langle b, 1\rangle)$

Then, $\langle a, 0.8\rangle$ is updated. Notice that this update has no effect because the T-structure no longer contains attributes of the form $\langle a, x\rangle$ waiting for update (the a-list is empty). After the update of $\langle b, 1\rangle$, the T-record $\langle 0, 1.1, \varphi_5 = \{b, {}^{0.1}/e\} \Rightarrow \{{}^{0.8}/c, d, {}^{0.6}/e\}\rangle$ of φ_5 is processed. We arrive to:

c: $[(1, \langle 1, 2, \varphi_6\rangle)]$

$ADD = (\langle c, 0.8\rangle, \langle d, 1\rangle, \langle e, 0.6\rangle)$
$NEWDEP = \{{}^{0.8}/a, b, {}^{0.8}/c, d, {}^{0.6}/e\}$
$UPDATE = (\langle c, 0.8\rangle, \langle d, 1\rangle, \langle e, 0.6\rangle)$

The algorithm updates $\langle c, 0.8\rangle$, $\langle d, 1\rangle$, $\langle e, 0.6\rangle$ however such updates are all without any effect because the d-list and e-list are already empty, and the c-list contains a single record with $1 \not\leq 0.8$ (see the condition at line 18 of the algorithm). Thus, the T-structure remains unchanged, $UPDATE$ is empty, and the procedure stops returning the value of $NEWDEP$ which is $\{{}^{0.8}/a, b, {}^{0.8}/c, d, {}^{0.6}/e\}$.

Example 2. In this example we demonstrate the role of the $WAITLIST$. Let T be a set of FAIs which consists of

ψ_1: $\{{}^{0.2}/a\} \Rightarrow \{{}^{0.6}/a, {}^{0.3}/c\}$,
ψ_2: $\{{}^{0.3}/c\} \Rightarrow \{{}^{0.2}/b\}$,
ψ_3: $\{{}^{0.6}/a, {}^{0.3}/c\} \Rightarrow \{b\}$,
ψ_4: $\{{}^{0.6}/a, b, {}^{0.3}/c\} \Rightarrow \{d\}$.

Moreover, we consider $M = \{{}^{0.3}/a\}$ and $PCLOSED = true$. After the initialization (line 14), we have $NEWDEP = \{{}^{0.3}/a\}$, $CARDND = 0.3$ ($f_{\mathbf{L}}$ is identity), $UPDATE = (\langle a, 0.3\rangle)$, $WAITLIST = ()$, and the T-structure is the following:

a: $[(0.2, \langle 1, 0.2, \psi_1\rangle), (0.6, \langle 3, 1.9, \psi_4\rangle, \langle 2, 0.9, \psi_3\rangle)]$
b: $[(1, \langle 3, 1.9, \psi_4\rangle)]$
c: $[(0.3, \langle 3, 1.9, \psi_4\rangle, \langle 2, 0.9, \psi_3\rangle, \langle 1, 0.3, \psi_2\rangle)]$

The computation continues with the update of $\langle a, 0.3\rangle$. During that, the T-record $\langle 1, 0.2, \psi_1\rangle$ will be updated to $\langle 0, 0.2, \psi_1\rangle$. Since $CARD[\psi_1] = 0.2 < 0.3 = CARDND$, the left-hand side of ψ_1 is strictly contained in $NEWDEP$, and the algorithm processes $\langle 0, 0.2, \psi_1 = \{{}^{0.2}/a\} \Rightarrow \{{}^{0.6}/a, {}^{0.3}/c\}\rangle$, i.e. we get to

a: $[(0.6, \langle 3, 1.9, \psi_4\rangle, \langle 2, 0.9, \psi_3\rangle)]$
b: $[(1, \langle 3, 1.9, \psi_4\rangle)]$
c: $[(0.3, \langle 3, 1.9, \psi_4\rangle, \langle 2, 0.9, \psi_3\rangle, \langle 1, 0.3, \psi_2\rangle)]$

$ADD = (\langle a, 0.6\rangle, \langle c, 0.3\rangle)$
$NEWDEP = \{{}^{0.6}/a, {}^{0.3}/c\}$
$CARDND = 0.9$
$UPDATE = (\langle a, 0.6\rangle, \langle c, 0.3\rangle)$

After the update of $\langle a, 0.6\rangle$, we have:

b: $[(1, \langle 2, 1.9, \psi_4\rangle)]$
c: $[(0.3, \langle 2, 1.9, \psi_4\rangle, \langle 1, 0.9, \psi_3\rangle, \langle 1, 0.3, \psi_2\rangle)]$

Then, the algorithm continues with updating $\langle c, 0.3\rangle$. The T-record $\langle 2, 1.9, \psi_4\rangle$ is updated to $\langle 1, 1.9, \psi_4\rangle$ and removed from the c-list. In the next step, the T-record $\langle 1, 0.9, \psi_3\rangle$ is updated to $\langle 0, 0.9, \psi_3\rangle$. At this point, we have $CARD[\psi_3] = 0.9 = CARDND$, i.e. we add fuzzy set $\{b\}$ of attributes (the right-hand side of

ψ_3) to the *WAITLIST*. Finally, $\langle 1, 0.3, \psi_2 \rangle$ is updated to $\langle 0, 0.3, \psi_2 \rangle$ which yields the following situation: the T-structure consists of b: $[(1, \langle 1, 1.9, \psi_4 \rangle)]$, $ADD = (\langle b, 0.2 \rangle)$, $NEWDEP = \{^{0.6}/a, ^{0.2}/b, ^{0.3}/c\}$, $CARDND = 1.1$, and $UPDATE = (\langle b, 0.2 \rangle)$. Since *ADD* is nonempty, the algorithm continues with flushing the *WAITLIST* (lines 25–33). After that, the new values are set to $NEWDEP = \{^{0.6}/a, b, ^{0.3}/c\}$, $CARDND = 1.9$, and $UPDATE = (\langle b, 0.2 \rangle, \langle b, 1 \rangle)$. The process continues with updating $\langle b, 0.2 \rangle$ (no effect) and $\langle b, 1 \rangle$. Here again, we are in a situation where $CARD[\psi_4] = 1.9 = CARDND$, i.e. $\{d\}$ is added to the *WAITLIST*, only this time, the computation ends because *UPDATE* is empty, i.e. $\{d\}$ will not be added to *NEWDEP*. Thus, the resulting value being returned is $\{^{0.6}/a, b, ^{0.3}/c\}$.

5 Conclusions

We have shown an extended version of the LinClosure algorithm, so-called Graded LinClosure (GLinClosure). Our algorithm can be used in case of graded as well as binary attributes. Even for binary attributes, GLinClosure is more versatile than the original LinClosure (it can be used to compute systems of pseudo-intents) but it has the same asymptotic complexity $O(n)$. Future research will focus on further algorithms for formal concept analysis of data with fuzzy attributes.

References

1. Bělohlávek R.: *Fuzzy Relational Systems: Foundations and Principles.* Kluwer, Academic/Plenum Publishers, New York, 2002.
2. Bělohlávek R., Chlupová M., Vychodil V.: Implications from data with fuzzy attributes. AISTA 2004 in Cooperation with the IEEE Computer Society Proceedings, 2004, 5 pages, ISBN 2–9599776–8–8.
3. Bělohlávek R., Vychodil V.: Reducing the size of fuzzy concept lattices by hedges. In: FUZZ-IEEE 2005, The IEEE International Conference on Fuzzy Systems, May 22–25, 2005, Reno (Nevada, USA), pp. 663–668 (proceedings on CD), abstract in printed proceedings, p. 44, ISBN 0–7803–9158–6.
4. Bělohlávek R., Vychodil V.: Fuzzy attribute logic: attribute implications, their validity, entailment, and non-redundant basis. In: Liu Y., Chen G., Ying M. (Eds.): *Fuzzy Logic, Soft Computing & Computational Intelligence: Eleventh International Fuzzy Systems Association World Congress* (Vol. I), 2005, pp. 622–627. Tsinghua University Press and Springer, ISBN 7–302–11377–7.
5. Bělohlávek R., Vychodil V.: Attribute implications in a fuzzy setting. In: Missaoui R., Schmid J. (Eds.): *ICFCA 2006*, LNAI **3874**, pp. 45–60, 2006.
6. Bělohlávek R., Vychodil V.: Functional dependencies of data tables over domains with similarity relations. In: *Proc. IICAI 2005*, pp. 2486–2504, ISBN 0–9727412–1–6.
7. Bělohlávek R., Vychodil V.: Data tables with similarity relations: functional dependencies, complete rules and non-redundant bases. In: Lee M. L., Tan K. L., Wuwongse V. (Eds.): *DASFAA 2006*, LNCS **3882**, pp. 644–658, 2006.

8. Bělohlávek R., Vychodil V.: Properties of models of fuzzy attribute implications (to appear in *Proc. SCIS & ISIS 2006,* Sep. 20–24, 2006, Tokyo, Japan).
9. Ganter B.: *Begriffe und Implikationen*, manuscript, 1998.
10. Ganter B.: Algorithmen zur formalen Begriffsanalyse. In: Ganter B., Wille R., Wolff K. E. (Hrsg.): *Beiträge zur Begriffsanalyse.* B. I. Wissenschaftsverlag, Mannheim, 1987, 241–254.
11. Ganter B., Wille R.: *Formal Concept Analysis. Mathematical Foundations.* Springer, Berlin, 1999.
12. Goguen J. A.: The logic of inexact concepts. *Synthese* **18**(1968-9), 325–373.
13. Guigues J.-L., Duquenne V.: Familles minimales d'implications informatives resultant d'un tableau de données binaires. *Math. Sci. Humaines* **95**(1986), 5–18.
14. Hájek P.: *Metamathematics of Fuzzy Logic.* Kluwer, Dordrecht, 1998.
15. Hájek P.: On very true. *Fuzzy Sets and Systems* **124**(2001), 329–333.
16. Klir G. J., Yuan B.: *Fuzzy Sets and Fuzzy Logic. Theory and Applications.* Prentice Hall, 1995.
17. Maier D.: *The Theory of Relational Databases.* Computer Science Press, Rockville, 1983.
18. Pavelka J.: On fuzzy logic I, II, III. *Z. Math. Logik Grundlagen Math.* **25**(1979), 45–52, 119–134, 447–464.
19. Pollandt S.: *Fuzzy Begriffe.* Springer-Verlag, Berlin/Heidelberg, 1997.
20. Takeuti G., Titani S.: Globalization of intuitionistic set theory. *Annals of Pure and Applied Logic* **33**(1987), 195–211.