# Relational Model of Data over Domains with Similarities: An Extension for Similarity Queries and Knowledge Extraction

Radim Belohlavek and Vilem Vychodil
Dept. Comp. Science, Palacky University, Tomkova 40, CZ-779 00, Olomouc, Czech Republic
e-mail: {radim.belohlavek, vilem.vychodil}@upol.cz

***Abstract*— We present an extension of Codd's relational model of data. Our extension is motivated by similarity-based querying. It consists in equipping each domain of attribute values with a similarity relation and in modifying the classical relational model in order to account for issues generated by adding similarities. As a counterpart to data tables over a set of domains of Codd's model, we introduce ranked data tables over domains with similarities. We present a relational algebra, and tuple and domain calculi for our model and prove their equivalence. An interesting point is that our relational algebra contains operations like $\mathrm{top}_k$ ($k$ best results matching a query). Then, we study functional dependencies extended by similarities, argue that they form a new type of data dependency not captured by the classical model, prove a completeness result w.r.t. Armstrong-like rules, describe non-redundant bases and provide an algorithm for computing the bases. In addition to that, we compare our model with other approaches and outline future research.**

## I. Introduction

### A. Motivation and outline of the paper

Codd's relational model of data is one of the most important contributions to computer science and perhaps the most important concept in data management ("A hundred years from now, I'm quite sure, database systems will still be based on Codd's relational foundation." [11, p. 1]). The main virtues of the model, i.e. logical and physical data independence, access flexibility, data integrity, etc., are mainly due to the reliance of the model on a simple yet powerful mathematical concept of a relation and first-order logic ("The relational approach really is rock solid, owing (once again) to its basis in mathematics and predicate logic." [11, p. 138]).

Since the inception of the relational model, there have been proposed various extensions of the model. Relevant to this paper are the extensions aiming at the capability of the relational model to deal with various forms of uncertainty. Management of uncertainty is one of the six currently most-important research directions proposed in the report from the Lowell debate by 25 senior database researchers [1]. In particular, it was pointed out in [1] that "...current DBMS have no facilities for either approximate data or imprecise queries."

Similarity, approximate matches, and the corresponding queries are the main motivations for our extension of the relational model. In particular, our primary concern is with the situation when domains are equipped with similarity relations, i.e. when it is desirable to consider degrees of similarity rather than only "equal" and "not equal". Such a concern comes primarily from the standpoint of information retrieval when considering similarity-based queries like "show all candidates with age about 30". Therefore, our attempt can be seen as extending the relational model by taking into account issues raised by similarity-based information retrieval. In addition to similarity-based queries, domains with similarity generate topics related to data mining. An example dealt with in more detail in our paper are functional dependencies which, in their extended version, represent new type of data dependencies like "similarity in values of attributes $A$ implies similarity in values of attributes $B$".

The main concept we deal with is that of a ranked data table (relation) over domains with similarities. This concept is our counterpart to the concept of a data table (relation) over domains of a classical relational model. The concept is illustrated in Tab. I. It consists of three parts: data table (relation), domain similarities, and ranking. The data table (right top table in Tab. I) coincides with a data table of a classical relational model. Domain similarities and ranking are what makes our model an extension of the classical model. The domain similarities (bottom part of Tab. I) assign degrees of similarity to pairs of values of the respective domain. For instance, a degree of similarity of "Computer Science" and "Computer Engineering" is 0.9 while a degree of similarity of "Computer Science" and "Electrical Engineering" is 0.6. The ranking assigns to each row (tuple) of the data table a degree of a scale bounded by 0 and 1 (left top table in Tab. I), e.g. 0.9 assigned to the tuple $\langle \mathrm{Chang}, 28, \mathrm{Accounting} \rangle$. The ranking allows us to view the ranked table as an answer to a similarity-based query (rank = degree to which a tuple matches a query). For instance, the ranked table of Tab. I can result as an answer to query "show all candidates with age about 30". In a data table representing stored data (i.e. prior to any querying), ranks of all tuples of the table are equal to 1. Therefore, the same way as tables in the classical relational model, ranked tables represent both stored data and outputs to queries. This is an important feature of our model.

TABLE I
RANKED DATA TABLE OVER DOMAINS WITH SIMILARITIES

| $\mathcal{D}(t)$ | *name* | *age* | *education* |
|---|---|---|---|
| 1.0 | Adams | 30 | Comput. Sci. |
| 1.0 | Black | 30 | Comput. Eng. |
| 0.9 | Chang | 28 | Accounting |
| 0.8 | Davis | 27 | Comput. Eng. |
| 0.4 | Enke | 36 | Electric. Eng. |
| 0.3 | Francis | 39 | Business |

$$n_1 \approx_n n_2 = \begin{cases} 1 & \text{if } n_1 = n_2 \\ 0 & \text{if } n_1 \neq n_2 \end{cases}$$

$$a_1 \approx_a a_2 = s_a(|a_1 - a_2|) \quad \text{with scaling } \; s_a : \mathbb{Z}^+ \to [0,1]$$

| $\approx_e$ | A | B | CE | CS | EE |
|---|---|---|---|---|---|
| A | 1 | .7 | | | |
| B | .7 | 1 | | | |
| CE | | | 1 | .9 | .6 |
| CS | | | .9 | 1 | .7 |
| EE | | | .6 | .7 | 1 |

We use fuzzy logic as our formal framework. In particular, we use a formal system of first-order fuzzy logic the same way as the system of first-order classical logic is used in the classical relational model. Using a formal system of first-order fuzzy logic enables us to utilize both the symbolical and the numerical character of fuzzy logic. That is, we work with first-order formulas which can be read in natural language and have thus a clear meaning (symbolical character). According to the rules of fuzzy logic, the formulas get assigned degrees, e.g. from $[0,1]$, which are being processed according to the rules of fuzzy logic (numerical character). This way, our model keeps the user-friendly symbolical character of the classical model and adds a quantitative layer which takes care of the management of uncertainty. This is an important distinction from other "fuzzy

0-7803-9788-6/06/$20.00 

approaches" to the relational model which, from our point of view, are often *ad-hoc*.

The paper is organized as follows. In Section I-B we review related approaches. Section I-C contains preliminaries from fuzzy logic. In Section II-A we introduce our model and basic related notions. Section II-B outlines relational algebra, tuple relational calculus, and domain relational calculus for our model and outlines their equivalence. Here, instead of going to technical details, we aim to emphasize the fact that the manipulative part of our model contains interesting operations which cannot be expressed in the classical model. We focus on non-classical issues like similarity-based selection, join, etc. An interesting point is that queries like $\text{top}_k$ (top $k$ answers to a similarity-based query, see e.g. [13]) can be made a natural part of our relational algebra. Section II-C deals with functional dependencies (FDs) in our extended model. Here, we present more technical details, primarily for the purpose of demonstrating the formal facet of our model. The main aim is to show that taking similarity into account, FDs describe an interesting type of data dependencies, tractable both theoretically and computationally. We show completeness results with Armstrong-like axioms, describe a non-redundant basis of all FDs of a ranked data table, and describe an algorithm for computing the bases. Section III outlines future research.

### B. Related approaches

Extensions of the classical relational model of data attempting to capture uncertainty and indeterminacy can be classified in a number of ways. One of them is according to the type of uncertainty. Before going to the approaches directly related to our paper we need to clearly distinguish our approach from probabilistic ones to prevent a confusion. The probabilistic models, see e.g. [12], [15], basically aim at modeling of probabilistic uncertainty of data which is very different from what we deal with, see e.g. [21]. Our model is deterministic; the only point where we depart from the classical model is that we use graded (fuzzy) predicates.

The first paper on a "fuzzy approach" to the relational model is [7]; [6] provides an overview with many references. We found over 100 contributions related to "fuzzy approach" to the relational model.

A main feature of almost all of the approaches is that they are *ad-hoc*. An *ad-hoc* choice of fuzzy logic connectives, the lack of symbolic level in the formalism, plus not clearly justifying why this and that is fuzzy rather than bivalent in a model has some important consequences. First of all are the impression of arbitrariness of the model; difficulties to handle the model theoretically and to arrive at some important features of the model; several concepts of the models are difficult to read and thus user-unfriendly; the impression of unjustified theorizing due to not explaining the meaning of fuzzy membership degrees. In our opinion, this is the main reason why the use of fuzzy logic in information systems did not spread wider that it did. In fact, most of the contributions were presented inside the fuzzy community. The primary reason of the above-described shortcomings is that an analogy of a clear relationship between a relational model and first-order fuzzy logic is missing in the approaches. This is partly because fully fledged logical calculi have not been developed until quite recently, see e.g. [16], [17].

Another feature of the approaches is that a consideration of computational tractability of the proposed concepts seems to be an exception rather than a rule.

On the other hand, several ideas including some of those presented in our paper were already discussed in the literature. For instance, the idea of considering domains with similarity relations goes back to [7]. The idea of assigning ranks to tuples appeared in [28] although with not quite a clear meaning of ranks (values of a "possibility distribution function", "fuzzy measure of association among a set of domain values" [28]). Quite several approaches exist in the literature on "fuzzy functional dependencies" and we comment on them in Section II-C. Another idea, which we do not consider here is the possibility for the tuples to contain also fuzzy sets of (or possibility distributions on) attribute values in addition to the attribute values themselves, [27] is perhaps the first paper developing this issue.

We comment on the relationships of our model to previous approaches later on; a detailed description will be presented elsewhere.

### C. Preliminaries

We use fuzzy logic to represent and manipulate truth degrees of propositions like "$u$ is similar to $v$". Moreover, we need to process (aggregate) the degrees. For instance, consider a query "show all candidates which are about 30 years old and a degree in specialization similar to Computer Science". According to Tab. I, Davis satisfies subqueries concerning age and education in degrees 0.8 and 0.9, respectively. Then, we combine the degrees using a fuzzy conjunction connective $\otimes$ to get a degree $0.8 \otimes 0.9$ to which Davis satisfies the conjunctive query.

When using fuzzy logic, we have to pick an appropriate scale $L$ of truth degrees (which serve e.g. as grades for evaluating similarity of two objets) and appropriate fuzzy logic connectives (conjunction, implication, etc.). Most of the existing fuzzy approaches to databases use the real interval $[0,1]$ and one particular couple of connectives on $[0,1]$. Our approach is different in that we do not say which particular scale and connectives we take. Rather, we postulate the required properties of the scale and of the connectives. Thus, we take an arbitrary partially-ordered scale $\langle L, \leq \rangle$ of truth degrees and require the existence of infima and suprema (for technical reasons, to be able to evaluate quantifiers). Furthermore, instead of taking one particular fuzzy conjunction $\otimes$ and fuzzy implication $\rightarrow$, we take any $\otimes$ and $\rightarrow$ which satisfy certain conditions. For, instance, our fuzzy conjunctions are order-preserving functions on $L$ satisfying some further requirements. This way, we obtain a structure $\mathbf{L} = \langle L, \leq, \otimes, \rightarrow, \ldots \rangle$ of truth degrees with logical connectives. Although more general than one particular choice of a scale and connectives, such an approach is easier to handle theoretically and supports the symbolical character of our model.

In what follows, we present technical details of the preliminaries; for further information, the reader is referred to [16], [17], [20].

For structures $\mathbf{L}$ of truth degrees, we use so-called complete residuated lattices, i.e. structures $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1 \rangle$ 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, i.e. $a \otimes b \leq c$ iff $a \leq b \rightarrow c$, for each $a, b, c \in L$. A truth-stressing hedge (shortly, a hedge) [17] on $\mathbf{L}$ is a unary operation $^* : L \rightarrow L$ satisfying (i) $1^* = 1$, (ii) $a^* \leq a$, (iii) $(a \rightarrow b)^* \leq a^* \rightarrow b^*$, (iv) $a^{**} = a^*$, for all $a, b \in L$. Elements $a$ of $L$ are called truth degrees. Hedge $^*$ is a (truth function of) logical connective "very true" and properties (i)–(iv) have natural interpretations, see [17].

A favorite choice of $\mathbf{L}$ is $L = [0,1]$ or a subchain of $[0,1]$. Examples of pairs of important pairs of adjoint operations are Łukasiewicz ($a \otimes b = \max(a+b-1, 0)$, $a \rightarrow b = \min(1-a+b, 1)$), and Gödel ($a \otimes b = \min(a,b)$, $a \rightarrow b = 1$ if $a \leq b$, $a \rightarrow b = b$ else). Two boundary cases of hedges are (i) identity, i.e. $a^* = a$ ($a \in L$); (ii) globalization: $1^* = 1$, and $a^* = 0$ ($a \neq 1$). Note that a special case of a complete residuated lattice with a hedge is a two-element Boolean algebra of classical (bivalent) logic.

Having $\mathbf{L}$, we define usual notions [16], [17], [20]: an $\mathbf{L}$-set (fuzzy set) $A$ in universe $U$ is a mapping $A : U \rightarrow L$, $A(u)$ being

interpreted as "the degree to which $u$ belongs to $A$". If $U$ is finite, we write $A = \{\ldots, {}^a/u, \ldots\}$ to denote that $A(u) = a \neq 0$. Let $\mathbf{L}^U$ denote the collection of all $\mathbf{L}$-sets in $U$. The operations with $\mathbf{L}$-sets are defined componentwise. Binary $\mathbf{L}$-relations (binary fuzzy relations) between $X$ and $Y$ can be thought of as $\mathbf{L}$-sets in the universe $X \times Y$. A fuzzy relation $E$ in $U$ is called reflexive if for each $u \in U$ we have $E(u,u) = 1$; symmetric if for each $u, v \in U$ we have $E(u,v) = E(v,u)$. A reflexive and symmetric fuzzy relation is called a similarity. We often denote a similarity by $\approx$ and use an infix notation, i.e. we write $(u \approx v)$ instead of $\approx(u,v)$. For fuzzy sets $A, B \in \mathbf{L}^U$, a degree $S(A,B)$ to which $A$ is a subset of $B$ is defined by $S(A,B) = \bigwedge_{u \in U}(A(u) \to B(u))$.

## II. Relational model over domains with similarities

### A. Basic concepts

In this section, we describe the basic concepts of our extended relational model. If not defined otherwise, we use the notions related to the relational model as defined in [23]. In our description, we concentrate on the issues related to domain similarities and table ranks. We use $Y$ for a set of attributes (attribute names) and denote the attributes by $y, y_1, \ldots$; $\mathbf{L}$ denotes a fixed structure of truth degrees and connectives.

*Definition 1:* A *ranked data table over domains with similarity relations* (with $Y$ and $\mathbf{L}$) is given by

- *domains*: for each $y \in Y$, $D_y$ is a non-empty set (domain of $y$, set of values of $y$);
- *similarities*: for each $y \in Y$, $\approx_y$ is a binary fuzzy relation (called similarity) in $D_y$ (i.e. a mapping $\approx_y : D_y \times D_y \to L$) which is reflexive (i.e. $u \approx_y u = 1$) and symmetric ($u \approx_y v = v \approx_y u$);
- *ranking*: for each tuple $t \in \times_{y \in Y} D_y$, there is a degree $\mathcal{D}(t) \in L$ (called rank of $t$ in $\mathcal{D}$) assigned to $t$.

*Remark 2:* (1) $\mathcal{D}$ can be seen as a table with rows and columns corresponding to tuples and attributes, like in Tab. I. By $t[y]$ we denote a value from $D_y$ of tuple $t$ on attribute $y$. We require that there is only a finite number of tuples which get assigned a non-zero degree (i.e. the corresponding table is finite). Clearly, if $L = \{0,1\}$ and if each $\approx_y$ is equality, the concept of a ranked data table with similarities coincides with that of a data table (relation) of a classical model.

(2) Formally, $\mathcal{D}$ is a fuzzy relation between domains $D_y$ ($y \in Y$). As mentioned above, $\mathcal{D}(t)$ is interpreted as a degree to which the tuple $t$ satisfies requirements posed by a query. We use "non-ranked table" if for each tuple $t$, $\mathcal{D}(t) = 0$ or $\mathcal{D}(t) = 1$. This accounts for tables representing strored data (prior to querying).

(3) Sometimes, we add additional requirements for $\approx_y$, e.g. transitivity w.r.t. a binary operation $\odot$ on $L$ or separability ($u \approx_y v = 1$ iff $u = v$). We are not concerned here with how the similarities are represented (we assume they can either be computed or, if $D_y$ is small, are stored).

(4) Ranked tables over domains with similarities appear in [28]. However, the authors consider only $[0,1]$ as a scale and no logical connectives.

### B. Relational algebra and calculus

*1) Relational algebra:* Relational algebra of the classical model is based on the calculus of classical relations. In the same spirit, since ranked tables are in fact fuzzy relations, our relational algebra is based on the calculus of fuzzy relations [16], [20]. Due to the limited scope, we present in detail only selected parts of our algebra and leave the rest in an outline. Details will be presented in a full version of the paper.

TABLE II
Illustration of similarity-based join

| $\mathcal{D}(t)$ | *position* | *education* |
|---|---|---|
| 1.0 | programmer | Comput. Sci. |
| 1.0 | syst. technician | Comput. Eng. |

| $\mathcal{D}(t)$ | *name* | *position* |
|---|---|---|
| 1.0 | Adams | programmer |
| 1.0 | Black | syst. technician |
| 0.9 | Adams | syst. technician |
| 0.9 | Black | programmer |

Our relational algebra is relative to $\mathbf{L}$ and manipulates ranked data tables with common $Y$, domains, and similarities. Operations of our relational algebra can be classified as follows.

*Counterparts to Boolean operations of classical model* Here, for any binary (and similar for other arities) operation $\odot$ with fuzzy relations, we define a corresponding operation (denoted again) $\odot$ which yields for any two ranked tables $\mathcal{D}_1$ and $\mathcal{D}_2$ (with common $Y$, domains, and similarities) a ranked table $\mathcal{D}$ assigning to any tuple $t$ a rank $\mathcal{D}(t)$ defined componentwise by

$$\mathcal{D}(t) = \mathcal{D}_1(t) \odot \mathcal{D}_2(t).$$

This accounts for operations based on $\wedge$, $\vee$, $\otimes$, $\to$, etc. (this way, we obtain our counterparts to intersection, union, etc.). Note that, one has to be careful when reducing operations to other operations. For instance, unlike classical case, De Morgan law is not available in fuzzy logic in general and, as a consequence, union cannot be expressed by intersection and complement.

*New operations based on calculus of fuzzy relations* The calculus of fuzzy relations contains operations which either have no counterparts with classical relations or the counterparts are trivial. An interesting example is a so-called $a$-cut of a fuzzy relation. For a ranked table $\mathcal{D}$ and a rank $a \in L$, an $a$-cut of $\mathcal{D}$ is a ranked table ${}^a\mathcal{D}$ defined by

$$[{}^a\mathcal{D}](t) = \begin{cases} 1 & \text{if } \mathcal{D}(t) \geq a, \\ 0 & \text{otherwise.} \end{cases}$$

That is, ${}^a\mathcal{D}$ is a non-ranked table which contains those tuples of $\mathcal{D}$ with ranks greater or equal to $a$. This is quite a natural operation for manipulation of ranked tables which allows the user to select only a part of a query result given by threshold $a$. Note that in combination with intersection, $a$-cut is able to keep the original ranks. Namely, we have $[\mathcal{D} \wedge {}^a\mathcal{D}](t) = \mathcal{D}(t)$ if $\mathcal{D}(t) \geq a$ and $= 0$ otherwise.

*Counterparts to selection, join, projection, etc.* These operation stem basically from the classical ones by taking into account similarity relations (or, in general fuzzy relations $\theta$ in place of classical comparators). For illustration, we consider a similarity-based join. For simplicity, consider a ranked table $\mathcal{D}_1$ from Tab. I (result to a query "... candidates with age about 30") and a ranked table $\mathcal{D}_2$ from Tab. II (top) describing open positions with required education. A similarity-based join $\mathcal{D}_1 \bowtie \mathcal{D}_2$ then describes possible job assignments. A rank $[\mathcal{D}_1 \bowtie \mathcal{D}_2](n,a,e,p)$ of tuple $\langle n,a,e,p \rangle$ in $\mathcal{D}_1 \bowtie \mathcal{D}_2$ is given by

$$\bigvee_{e_1,e_2}(\mathcal{D}_1(n,a,e_1) \otimes (e_1 \approx_e e) \otimes (e \approx_e e_2) \otimes \mathcal{D}_2(p,e_2))$$

where $e_1, e_2$ range over the domain corresponding to *education*. That is, the join runs not only over equal values but also over similar values at the cost of decreasing the value of the resulting tuples by degrees of similarity. The bottom table of Tab. II shows a result of a 0.9-cut of $\mathcal{D}_1 \bowtie \mathcal{D}_2$ projected to *name* and *position*.

*Further operations* ($\text{top}_k$ *etc.*) Here, we put operations interesting from the point of information retrieval which cannot be accounted for in classical model. As an example, consider $\text{top}_k$ which gained a considerable interest recently, see [13], [14] and also [18]. We define $\text{top}_k(\mathcal{D})$ to contain the first $k$ tuples (according to rank ordering) of $\mathcal{D}$ with their ranks (if there are less than $k$ ranks in $\mathcal{D}$ then $\text{top}_k(\mathcal{D}) = \mathcal{D}$; and $\text{top}_k(\mathcal{D})$) includes also the tuples with rank equal to the rank of the $k$-th tuple). Note that $\text{top}_k$ is a part of a query language described in [26].

*2) Tuple and domain relational calculi:* The tuple calculus of classical model is based on classical predicate logic. In the same spirit (here again, as with relational algebra), our tuple calculus is based on fuzzy predicate logic. It is important for our purpose that predicate fuzzy logic(s) are developed nowadays and that they are in a relationship to the calculus of fuzzy relations similar to the relationship of classical predicate logic to the calculus of classical relations. Expressions of our tuple calculus are of the form

$$\{x(R) \mid f(x)\}$$

with the usual meaning of the components ($x$ the only free variable in a legal formula $f$, $R$ a set of attributes). Formulas $f(x)$ are built from atoms using symbols of connectives of fuzzy logic in the usual way. In addition to this, atoms include truth constants $a \in L$, and we need a unary connective $\Delta$ (Baaz's delta [17]). We have also non-standard quantifiers [17] in our language like $Q_{<k}$ ("less than $k$") with $(Q_{<k}x)f(x)$ having truth degree 1 if the number of tuples for which $f(x)$ evaluates to a non-zero degree is less than $k$ and having truth degree 0 otherwise. Due to inclusion of $Q_{<k}$, tuple calculus has expressions equivalent to $\text{top}_k$, one of them being a formula

$$\mathcal{D}(x) \wedge (Q_{<k}y)(\neg\Delta(\mathcal{D}(y) \to \mathcal{D}(x)) \wedge \Delta(\mathcal{D}(x) \to \mathcal{D}(y))).$$

The situation is similar for a domain relational calculus.

Taking appropriate care of the details, one can obtain the following theorem (the details and proof will be presented in a full version).

*Theorem 3 (equivalence theorem):* Our relational algebra, domain calculus, and tuple calculus are mutually equivalent. ■

That is, for any expression $E_A$ of our relational algebra there is an expression $E_D$ of our domain calculus such that for any state of a database $d$, the ranked tables $E_A(d)$ and $E_D(d)$, to which $E_A$ and $E_D$ evaluate, coincide and *vice versa* (and the same for the other cases).

*Remark 4:* Previous approaches either consider only similarities [9] or only ranks [29] but not both. Most importantly, our approach provides more expressive power (inclusing e.g. $\text{top}_k$) and a firm connection to predicate fuzzy logic due to which both the relational algebra and calculi are open for further extensions (e.g. by other non-standard quantifiers, aggregation operators, etc.). [22] presents an interesting framework different from our one but with similar aims.

### C. Functional dependencies

Functional dependencies (FDs) describe a particular form of relationship. FDs are traditionally used for issues related to database design [23] and for obtaining information from data [24]. We are going to argue that in our setting, FDs extended by taking into account the domain similarities (1) provide us with a new type of data dependency; (2) leave many of the previous approaches to fuzzy FD particular cases; (3) are tractable both theoretically and computationally in an analogous way as with classical FDs. Claim (3) is particularly important since most of the previous approaches to fuzzy FD are confined to definitions and illustrative examples.

*1) Definition and related approaches:* In our setting, a (fuzzy) FD is a formula $A \Rightarrow B$ where $A$ and $B$ are fuzzy sets of attributes ($A, B \in \mathbf{L}^Y$). We first present a definition of validity of $A \Rightarrow B$ in a ranked data table $\mathcal{D}$ and then add comments.

*Definition 5:* For a ranked data table $\mathcal{D}$, tuples $t_1, t_2$ and a fuzzy set $C \in \mathbf{L}^Y$ of attributes, we introduce a *degree* $t_1(C) \approx_{\mathcal{D}} t_2(C)$ *to which* $t_1$ *and* $t_2$ *have similar values on attributes from* $C$ by

$$t_1(C) \approx_{\mathcal{D}} t_2(C) = \\ = (\mathcal{D}(t_1) \otimes \mathcal{D}(t_2)) \to \textstyle\bigwedge_{y \in Y} (C(y) \to (t_1[y] \approx_y t_2[y])). \quad (1)$$

A *degree* $||A \Rightarrow B||_{\mathcal{D}}$ *to which a FD* $A \Rightarrow B$ *is true in* $\mathcal{D}$ is defined by

$$||A \Rightarrow B||_{\mathcal{D}} = \textstyle\bigwedge_{t_1,t_2} \big((t_1(A) \approx_{\mathcal{D}} t_2(A))^* \to (t_1(B) \approx_{\mathcal{D}} t_2(B))\big). \quad (2)$$

*Remark 6:* (1) By basic rules of semantics of predicate fuzzy logic [17], $t_1(C) \approx_{\mathcal{D}} t_2(C)$ is just the truth degree of a formula "if $t_1, t_2$ are from $\mathcal{D}$ then for each attribute $y$ from $C$, $t_1$ and $t_2$ have similar values on $y$".

(2) Therefore, using predicate fuzzy logic again, $||A \Rightarrow B||_{\mathcal{D}}$ is a truth degree of a formula "*for any tuples* $t_1, t_2$*: if* $t_1$ *and* $t_2$ *have similar values on attributes from* $A$ *then* $t_1$ *and* $t_2$ *have similar values on attributes from* $B$". Note that due to our adherence to predicate fuzzy logic, the meaning of $A \Rightarrow B$ is given by a simple formula which we just described in natural language. Note that, in fact, the antecedent in formula (2) is modified by a hedge $^*$. This has technical reasons not discussed in detail here (note only that setting $^*$ to globalization or identity enables as to have some of the previous approaches as particular cases of our ones).

(3) Note also that $||A \Rightarrow B||_{\mathcal{D}}$ is a truth degree from our scale $L$, not necessarily being 0 or 1. That is, our FDs may be true to a degree, e.g., 0.9 (approximately true) which is natural when considering approximate concepts like similarity. The particular value of $||A \Rightarrow B||_{\mathcal{D}}$ depends on our choice of the scale and the connectives. For illustration, if the ranks in $\mathcal{D}$ are all 0 or 1 and $^*$ is globalization then for any choice of a scale $L$ and connectives $\otimes, \to$ we have that $||A \Rightarrow B||_{\mathcal{D}} = 1$ ($A \Rightarrow B$ is fully true in $\mathcal{D}$) means that for any tuples $t_1, t_2$ from $\mathcal{D}$: if $A(y) \leq (t_1[y] \approx_y t_2[y])$ for any attribute $y \in Y$ then $B(y) \leq (t_1[y] \approx_y t_2[y])$ for any attribute $y \in Y$. This also shows that degrees $A(y)$ and $B(y)$ serve basically as similarity thresholds.

(4) Compared to previous approaches to fuzzy FDs, see [4], [5], [10], [19], [27], [28], [30] for a representative sample, the following are the main distinctions of our approach. (i) Previous approaches use ordinary sets $A$ and $B$ in a fuzzy FD $A \Rightarrow B$ which is a special case in our setting since we allow fuzzy sets for $A$ and $B$. This allows us to express similarity thresholds (see above) which means a greater but still natural expressive power. (ii) Previous approaches do not consider approximate validity of FDs and related notions (like degree of entailment). (iii) Previous approaches use one particular scale, namely $[0,1]$, and one particular choice of connectives and do not consider relationship to predicate fuzzy logic. As a result, a simple natural language description of the meaning of fuzzy FD is missing. Furthermore, most of the previous approaches are a particular case of our approach. Due to the limited scope of our paper, we present a detailed comparison elsewhere.

*2) Armstrong-like axioms and completeness:* In this section, we present two kinds of complete axiomatization of our FDs by means of Armstrong-like axioms. Instead of proceeding directly (which is possible), we follow a shorter path by using results from [2] where complete axiomatizations were shown for fuzzy FDs evaluated over non-ranked data tables (i.e., ranks equal to 1 or 0 in our setting).

First, we need the following concepts. For a set $T$ of fuzzy FDs, let $\text{Mod}(T)$ be a set of all ranked data tables with similarities in which

each FD from $T$ is true in degree 1, i.e. $\mathrm{Mod}(T) = \{\mathcal{D} |$ for each $A \Rightarrow B \in T : ||A \Rightarrow B||_{\mathcal{D}} = 1\}$. $\mathcal{D} \in \mathrm{Mod}(T)$ are called models of $T$. A *degree* $||A \Rightarrow B||_T$ *to which* $A \Rightarrow B$ *semantically follows from* $T$ is defined by

$$||A \Rightarrow B||_T = \bigwedge_{\mathcal{D} \in \mathrm{Mod}(T)} ||A \Rightarrow B||_{\mathcal{D}}$$

where the infimum ranges over all models of $T$. Note that according to standard rules of fuzzy logic, $||A \Rightarrow B||_T$ is a degree to which it is true that $A \Rightarrow B$ is true in each model of $T$. The following is our "reduction result" which provides a connection to [2].

*Theorem 7:* For any set $T$ of FDs, we have

$$||A \Rightarrow B||_T = \bigwedge_{\mathcal{D} \in \mathcal{M}(T)} ||A \Rightarrow B||_{\mathcal{D}}, \tag{3}$$

where $\mathcal{M}(T) = \{\mathcal{D} \in \mathrm{Mod}(T) \,|\, \text{for each } t : \mathcal{D}(t) \in \{0,1\}\}$.

*Proof:* "$\leq$" is trivial because $\mathcal{M}(T) \subseteq \mathrm{Mod}(T)$.

"$\geq$": It suffices to check that for each data table $\mathcal{D} \in \mathrm{Mod}(T)$ there is $\mathcal{D}' \in \mathcal{M}(T)$ such that $||A \Rightarrow B||_{\mathcal{D}} = ||A \Rightarrow B||_{\mathcal{D}'}$. Take $\mathcal{D} \in \mathrm{Mod}(T)$. Consider a set $I$ for which we have $|I| = |\mathrm{Supp}(\mathcal{D})|$, i.e. the cardinality of $I$ is the same as the cardinality of the set of tuples which belong to $\mathcal{D}$ to a nonzero degree. $I$ will be used as a set of identifiers of tuples from $\mathcal{D}$. Fix any bijective mapping $f : I \to \mathrm{Supp}(\mathcal{D})$. We now define $\mathcal{D}'$ as follows:

- each domain $D'_y$ equals to $I$, i.e. we put $D'_y = I$;
- similarities $\approx'_y$ on domains $D'_y$ are defined by:
  $i \approx'_y j = (\mathcal{D}(f(i)) \otimes \mathcal{D}(f(j))) \to ((f(i))[y] \approx_y (f(j))[y])$
  for each $i, j \in D'_y$, $y \in Y$;
- for each $t \in \mathrm{Supp}(\mathcal{D})$, $\mathcal{D}'$ fully contains a tuple $t'$ (i.e., $\mathcal{D}(t') = 1$) such that $t'[y] = f^{-1}(t)$ $(y \in Y)$.

We can show that $||A \Rightarrow B||_{\mathcal{D}} = ||A \Rightarrow B||'_{\mathcal{D}}$. Indeed, observe that for any $t'_1, t'_2 \in \mathcal{D}'$, for the corresponding $t_1, t_2 \in \mathrm{Supp}(\mathcal{D})$, and for each fuzzy set $C$ of attributes we have

$$\begin{aligned}
&t'_1(C) \approx_{\mathcal{D}'} t'_2(C) = \\
&\quad = (\mathcal{D}(t'_1) \otimes \mathcal{D}(t'_2)) \to \textstyle\bigwedge_{y \in Y} (C(y) \to (t'_1[y] \approx'_y t'_2[y])) = \\
&\quad = 1 \to \textstyle\bigwedge_{y \in Y} (C(y) \to (t'_1[y] \approx'_y t'_2[y])) = \\
&\quad = \textstyle\bigwedge_{y \in Y} (C(y) \to (t'_1[y] \approx'_y t'_2[y])) = \\
&\quad = \textstyle\bigwedge_{y \in Y} (C(y) \to ((\mathcal{D}(t_1) \otimes \mathcal{D}(t_2)) \to (t_1[y] \approx_y t_2[y]))) = \\
&\quad = (\mathcal{D}(t_1) \otimes \mathcal{D}(t_2)) \to \textstyle\bigwedge_{y \in Y} (C(y) \to (t_1[y] \approx_y t_2[y])) = \\
&\quad = t_1(C) \approx_{\mathcal{D}} t_2(C).
\end{aligned}$$

The rest follows from definition of $||\cdots||_{\mathcal{D}}$. ∎

Our axiomatic system consists of the following deduction rules.

(Ax) infer $A \cup B \Rightarrow A$,

(Cut) from $A \Rightarrow B$ and $B \cup C \Rightarrow D$ infer $A \cup C \Rightarrow D$,

(Mul) from $A \Rightarrow B$ infer $c^* \otimes A \Rightarrow c^* \otimes B$

for each $A, B, C, D \in \mathbf{L}^Y$, and $c \in L$. Here, $c^* \otimes A \in \mathbf{L}^Y$ is defined by $(c^* \otimes A)(y) = c^* \otimes A(y)$. As usual, $A \Rightarrow B$ is called *provable* from a set $T$ of FDs, written $T \vdash A \Rightarrow B$, if there is a sequence $\varphi_1, \ldots, \varphi_n$ of FDs such that $\varphi_n$ is $A \Rightarrow B$ and for each $\varphi_i$ we either have $\varphi_i \in T$ or $\varphi_i$ is inferred (in one step) from some of the preceding FDs (i.e., $\varphi_1, \ldots, \varphi_{i-1}$) using some deduction rule (Ax)–(Mul).

*Theorem 8 (completeness):* Let $T$ be a set of FDs, $\mathbf{L}$ and $Y$ be finite. For each $A \Rightarrow B$ we have

$$T \vdash A \Rightarrow B \quad \text{iff} \quad ||A \Rightarrow B||_T = 1.$$

*Proof:* Sketch of the proof: The "$\Rightarrow$"-part of the claim (soundness) is routine to check by induction on length of a proof. Hint: observe that (Ax) is fully true in each ranked data table, and (Cut) and (Mul) infer fully true FDs (in $\mathcal{D}$) from fully true FDs (in $\mathcal{D}$). Due to Theorem 7, we can restrict ourselves only to models from $\mathcal{M}(T)$.

In order to show the "$\Leftarrow$"-part of the claim, it suffices to show that $T \nvdash A \Rightarrow B$ implies $||A \Rightarrow B||_T \neq 1$. Assuming $T \nvdash A \Rightarrow B$, we find a ranked data table $\mathcal{D} \in \mathrm{Mod}(T)$ such that $||A \Rightarrow B||_{\mathcal{D}} \neq 1$. Consider a system $S_A = \{C \in \mathbf{L}^Y \,|\, T \vdash A \Rightarrow C\}$ of fuzzy sets of attributes. $S_A$ has a greatest element (this follows from finiteness of $Y$, $\mathbf{L}$, and since $A \in S_A$). Denote the greatest element of $S_A$ by $A^+$. Take any data table $\mathcal{D}$ which consists of two tuples $t_1, t_2$ such that $\mathcal{D}(t_1) = \mathcal{D}(t_2) = 1$, and $t_1[y] \approx_y t_2[y] = A^+(y)$ $(y \in Y)$. Using the fact $T \nvdash A \Rightarrow B$, one can check that $\mathcal{D} \in \mathcal{M}(T)$ (i.e., $\mathcal{D}$ is a model of $T$), and that $||A \Rightarrow B||_{\mathcal{D}} \neq 1$. Details are postponed to the full version of the paper. ∎

Theorem 8 says that for a set $T$ and a FD $A \Rightarrow B$, $A \Rightarrow B$ is fully entailed by $T$ (i.e., in degree 1) iff $A \Rightarrow B$ is provable from $T$. Next, we extend this result to graded completeness (see [17] for details on this concept) to account for a general degree of entailment (i.e., other than 1). We proceed by reduction to the above-concept of provability (this is a luck in our setting which is not available in other situations; however, reasoning directly is also possible). For a set $T$ of FDs and for $A \Rightarrow B$ define a *degree* $|A \Rightarrow B|_T \in L$ *to which* $A \Rightarrow B$ *is provable from* $T$ by

$$|A \Rightarrow B|_T = \bigvee \{c \in L \,|\, T \vdash A \Rightarrow c \otimes B\}.$$

Then, the concept of a degree of provability coincides with that of a degree of semantic entailment.

*Theorem 9 (graded completeness):* Let $\mathbf{L}$ and $Y$ be finite. Then for every $T$ and $A \Rightarrow B$ we have

$$|A \Rightarrow B|_T = ||A \Rightarrow B||_T.$$

*Proof:* Due to Theorem 8, it is enough to show that $||A \Rightarrow B||_T = \bigvee \{c \in L \,|\, ||A \Rightarrow c \otimes B||_T = 1\}$, which is indeed true (follows from properties of a residuated implication, details are omitted). ∎

The idea of graded completeness generalizes also for fuzzy sets $T$ of formulas (i.e. for reasoning from partially true premises). Details are omitted due to the limited scope.

*Remark 10:* (1) The presented results generalize well-known results on completeness of Armstrong axioms [23]. Our results "became" the classical ones if we take a two-element Boolean algebra for our scale of truth degrees with connectives (classical conjunction and implication for $\otimes$ and $\to$, and identity for $^*$).

(2) Compared to the previous approaches to axiomatization of fuzzy FDs, note first that we deal with more general notion of a FD (see above). Our axioms differ from those reported in the literature on fuzzy FDs in that all of the reported results use only a set of classical Armstrong axioms. Contrary to that, we need a new rule (Mul). Furthermore, the previous approaches did not consider entailment in degrees and thus there are no attempts reported on graded completeness.

(3) Note also that in the previous approaches, the authors prove their completeness results directly. Since they use only the classical Armstrong axioms, it might be interesting to see if their completeness results follow from the completeness of classical FDs. This is, indeed, the case; we omit details (sketch for the case presented in [28]: for each data table with similarities one can construct a classical data table such that the tables have the same true FDs; the result then follows by a simple reasoning on semantic entailment).

*3) Computing non-redundant basis:* In this section, we focus on non-redundant bases of FDs of ranked data tables, i.e. minimal sets $T$ of FDs which are fully true in a given ranked table $\mathcal{D}$ and such that any other FD true in $\mathcal{D}$ follows semantically from $T$ in degree 1. Non-redundant bases are therefore minimal sets of FDs

which convey information about all fully true FDs in the table. The interest in obtaining non-redundant bases is basically twofold. First, from the point of view of knowledge extraction, a ranked data table $\mathcal{D}$ represents an answer to a similarity-based query. A non-redundant basis of $\mathcal{D}$ thus represents an additional information to the query which describes all dependencies satisfied by the result to the query. Second, as in the classical case, non-redundant sets of FDs are important in considerations concerning data redundancy and normalization (this applies particularly to non-ranked tables).

Computational aspects of fuzzy approaches to FDs are scarcely discussed in the literature and [30] seems to be an exception. However, since the aim in [30] is different from computing non-redundant bases, we do not discuss it here (in [30], the authors compute *all* FDs satisfying some non-triviality conditions).

In what follows, we make use of [3] where the problem of description and computation of a particular non-redundant basis was solved for non-ranked data tables with similarities (i.e., all ranks equal to 1 or 0). We present a couple of results which make it possible to apply results from [3] to the problem of non-redundant bases of ranked data tables. This way, we extend the results and methods of [3] to account for the more general case of ranked data tables.

Let thus $\mathcal{D}$ be a ranked data table with similarities.

*Definition 11:* A set $T$ of FDs is *complete in* $\mathcal{D}$ if, for each $A \Rightarrow B$, $||A \Rightarrow B||_T = ||A \Rightarrow B||_{\mathcal{D}}$. Moreover, if $T$ is complete in $\mathcal{D}$ and no proper subset of $T$ is complete in $\mathcal{D}$, we call $T$ a *non-redundant basis of* $\mathcal{D}$. $T$ is called a *minimal basis of* $\mathcal{D}$ if $T$ is complete in $\mathcal{D}$ and for each $T'$ which is complete in $\mathcal{D}$ we have $|T| \leq |T'|$.

We now proceed in two steps: First, we define a special closure operator $\mathrm{C}_{\mathcal{D}}$ which assigns to any fuzzy set $A$ of attributes its closure $\mathrm{C}_{\mathcal{D}}(A)$, which is again a fuzzy set of attributes, so that $T = \{A \Rightarrow \mathrm{C}_{\mathcal{D}}(A) \,|\, A \in \mathbf{L}^Y\}$ is complete in $\mathcal{D}$. Second, we describe a "small subset" of $T$ which is non-redundant (and minimal in size in some important cases) and computationally tractable. The first part of the procedure (description of $\mathrm{C}_{\mathcal{D}}$) is treated in more detail because it is a non-trivial extension of previous results. The second part (selecting a non-redundant subset) follows the same procedure as in [3], so we give only a hint.

*Definition 12:* For a ranked data table $\mathcal{D}$ over attributes $Y$ define an operator $\mathrm{C}_{\mathcal{D}} \colon \mathbf{L}^Y \to \mathbf{L}^Y$ by

$$(\mathrm{C}_{\mathcal{D}}(A))(y) = \\ = \bigwedge_{t,t'} ((\mathcal{D}(t) \otimes \mathcal{D}(t') \otimes (t(A) \approx t'(A))^*) \to (t[y] \approx_y t'[y])).$$

Observe that the tuples $t$ for which $\mathcal{D}(t) = 0$ can be disregarded in the formula for $\mathrm{C}_{\mathcal{D}}$.

*Theorem 13:* For each $\mathcal{D}$, $\mathrm{C}_{\mathcal{D}}$ is a closure operator, and $T = \{A \Rightarrow \mathrm{C}_{\mathcal{D}}(A) \,|\, A \in \mathbf{L}^Y\}$ is complete in $\mathcal{D}$.

*Proof:* Sketch of the proof: Using properties of residuated lattices and hedges, one can show that $\mathrm{C}_{\mathcal{D}}$ is a closure operator (the proof is technically involved and omitted due to the lack of space). For the second part, it suffices to show that models of $T$ entail exactly the same FDs as $\mathcal{D}$ does. This can be proved by showing that $\mathcal{D}$ is a model of $T$ (which follows from the definition of $\mathrm{C}_{\mathcal{D}}$) and that each model of $T$ entails all FDs which are entailed by $\mathcal{D}$ (hint: suppose some model of $T$ does not entail $A \Rightarrow B$, from which one gets $A \Rightarrow B \notin T$, i.e. $B \not\subseteq \mathrm{C}_{\mathcal{D}}(A)$, which further gives $||A \Rightarrow B||_{\mathcal{D}} \neq 1$ by definition of $\mathrm{C}_{\mathcal{D}}$). ■

We now focus on finding a non-redundant basis of $\mathcal{D}$ which is a subset of the set $T$ described in Theorem 13. Similarly as in [3], we take advantage of the technical concept of a system of pseudo-closed fuzzy sets of attributes. In the present setting of tables with truth weighted tuples, we define the notion as follows. Given $\mathcal{D}$, a collection $\mathcal{P} \subseteq \mathbf{L}^Y$ of fuzzy sets of attributes is called a *system of pseudo-closed fuzzy sets w.r.t.* $\mathcal{D}$ if, for each $P \in \mathbf{L}^Y$, we have:

$$P \in \mathcal{P} \quad \text{iff} \quad P \neq \mathrm{C}_{\mathcal{D}}(P) \text{ and for each } Q \in \mathcal{P} \\ \text{such that } Q \neq P\colon\ S(Q,P)^* \leq S(\mathrm{C}_{\mathcal{D}}(Q), P),$$

where "$S(\cdot,\cdot)$" denote degrees of subsethood, see Section I-C. Each $P \in \mathcal{P}$ is then called a *pseudo-closed fuzzy set of attributes*. One can prove the following assertion (the proof is postponed to the full version of the paper).

*Theorem 14:* If $\mathcal{P}$ is a system of pseudo-closed fuzzy sets w.r.t. $\mathcal{D}$, then $T = \{P \Rightarrow \mathrm{C}_{\mathcal{D}}(P) \,|\, P \in \mathcal{P}\}$ is a non-redundant basis of $\mathcal{D}$. If $^*$ is globalization, then $T$ is a minimal basis of $\mathcal{D}$. ■

*Remark 15:* The non-redundant basis $T$ of a ranked table $\mathcal{D}$ of Theorem 14 can be efficiently computed by an algorithm with polynomial time delay. Namely, the systems of pseudo-closed fuzzy sets introduced in our paper satisfy the requirements of the algorithms proposed for non-ranked data tables [3, Theorem 5, Algorithm 1]. We omit the presentation of the resulting algorithm due to space limitations.

EXAMPLE. We now present an example of a non-redundant basis of a ranked table. We consider a linear scale of 11 truth

TABLE III
ILLUSTRATIVE DATA TABLE: POWER CONSUMPTION OF COUNTRIES WITH VERY LARGE POPULATIONS

| $\mathcal{D}(t)$ | *country* | *coal* | *air* | *water* | *nuclear* |
|---|---|---|---|---|---|
| 1.0 | China | 498.0 | 246 | 196 | 34.6 |
| 1.0 | India | 154.3 | 1032 | 75 | 26.8 |
| 0.6 | USA | 570.7 | 2533 | 330 | 753.9 |
| 0.3 | Russia | 115.8 | 54 | 157 | 122.5 |
| 0.3 | Japan | 0.0 | 120 | 90 | 293.8 |
| 0.2 | Germany | 56.4 | 3817 | 50 | 161.2 |
| 0.2 | UK | 19.5 | 350 | 8 | 81.7 |
| 0.2 | France | 0.0 | 63 | 62 | 394.4 |
| 0.1 | Spain | 10.9 | 1180 | 11 | 58.9 |

TABLE IV
ILLUSTRATIVE DATA TABLE: PARTICULAR SIMILARITY RELATIONS ON DOMAINS

| $\approx_c$ | Cn | In | US | Ru | Jp | Ge | Fr | UK | Sp |
|---|---|---|---|---|---|---|---|---|---|
| Cn | 1 | | .3 | | | | | | |
| In | | 1 | | .6 | | | | | |
| US | .3 | | 1 | | | | | | |
| Ru | | .6 | | 1 | | .4 | | | |
| Jp | | | | | 1 | .4 | 1 | .8 | .9 |
| Ge | | | | .4 | .4 | 1 | .4 | .7 | .6 |
| Fr | | | | | 1 | .4 | 1 | .8 | .9 |
| UK | | | | | .8 | .7 | .8 | 1 | 1 |
| Sp | | | | | .9 | .6 | .9 | 1 | 1 |

| $\approx_a$ | Cn | In | US | Ru | Jp | Ge | Fr | UK | Sp |
|---|---|---|---|---|---|---|---|---|---|
| Cn | 1 | .5 | | .9 | .9 | | .9 | .9 | .4 |
| In | .5 | 1 | | .3 | .4 | | .3 | .5 | .9 |
| US | | | 1 | | | .1 | | | .1 |
| Ru | .9 | .3 | | 1 | 1 | | 1 | .8 | .2 |
| Jp | .9 | .4 | | 1 | 1 | | 1 | .9 | .3 |
| Ge | | | .1 | | | 1 | | | |
| Fr | .9 | .3 | | 1 | 1 | | 1 | .8 | .2 |
| UK | .9 | .5 | | .8 | .9 | | .8 | 1 | .4 |
| Sp | .4 | .9 | .1 | .2 | .3 | | .2 | .4 | 1 |

| $\approx_w$ | Cn | In | US | Ru | Jp | Ge | Fr | UK | Sp |
|---|---|---|---|---|---|---|---|---|---|
| Cn | 1 | | | .6 | | | | | |
| In | | 1 | | | 1 | .9 | 1 | .2 | .2 |
| US | | | 1 | | | | | | |
| Ru | .6 | | | 1 | .2 | | | | |
| Jp | | 1 | | .2 | 1 | .6 | .8 | | |
| Ge | | .9 | | | .6 | 1 | 1 | .6 | .6 |
| Fr | | 1 | | | .8 | 1 | 1 | .4 | .4 |
| UK | | .2 | | | | .6 | .4 | 1 | 1 |
| Sp | | .2 | | | | .6 | .4 | 1 | 1 |

| $\approx_n$ | Cn | In | US | Ru | Jp | Ge | Fr | UK | Sp |
|---|---|---|---|---|---|---|---|---|---|
| Cn | 1 | 1 | | .7 | | .4 | | 1 | 1 |
| In | 1 | 1 | | .6 | | .4 | | .9 | 1 |
| US | | | 1 | | | | | | |
| Ru | .7 | .6 | | 1 | .1 | 1 | | 1 | .9 |
| Jp | | | | .1 | 1 | .4 | .6 | | |
| Ge | .4 | .4 | | 1 | .4 | 1 | | .8 | .6 |
| Fr | | | | | .6 | | 1 | | |
| UK | 1 | .9 | | 1 | | .8 | | 1 | 1 |
| Sp | 1 | 1 | | .9 | | .6 | | 1 | 1 |

degrees 0 (falsity) $< 0.1 < 0.2 < \cdots < 1$ (full truth) equipped with Łukasiewicz connectives [17] and globalization. Table III describes power consumption of selected countries. The attributes denote name of the county, mass of coal (megatons) produced for power purposes, electricity (MW) produced by air power-plants, electricity ($10^3$ MW) produced by water power-plants, electricity ($10^{12}$ MW) produced by nuclear power-plants. For simplicity, we use names as tuples' identifiers of tuples instead of values of attributes.

Introducing similarity relations enables us to gain more information from the data. Let our similarities be given by Table IV. Our purpose is neither to study methods of specifying suitable similarities for particular data nor argue that our choice of similarities is "the best one"–this is a matter connected with particular problem domain (geography and economy, in this particular example) and should be left to experts in the areas.

Suppose first that a rank of each tuple in Table III is 1. Then the minimal basis of such a data table (with the underlying similarity relations) consists of 56 FDs.

If ranks of tuples are as given by the $\mathcal{D}(t)$-column of Table III, then the table can be seen as a result of a query "select power consumption of countries with *very large populations*". Intuitively, one may expect the the minimal basis of such a table would be smaller than the basis of the latter one because now several tuples (like Spain, France, ...) have a low rank (the populations are rather small). Indeed, the minimal basis given by the algorithm described in previous section is (after reduction of left-hand and right-hand sides of FDs) the following:

$$\{c, {}^{0.8}/w\} \Rightarrow \{w\}, \qquad \{{}^{0.1}/c\} \Rightarrow \{{}^{0.4}/c, {}^{0.4}/w\},$$
$$\{{}^{0.9}/c, {}^{0.8}/w\} \Rightarrow \{{}^{0.9}/w\}, \qquad \{{}^{0.6}/a\} \Rightarrow \{{}^{0.7}/c, {}^{0.8}/a, {}^{0.7}/w\},$$
$$\{{}^{0.9}/c\} \Rightarrow \{a, n\}, \qquad \{{}^{0.9}/n\} \Rightarrow \{n\},$$
$$\{{}^{0.8}/c\} \Rightarrow \{{}^{0.8}/a, {}^{0.7}/w, {}^{0.8}/n\}, \qquad \{{}^{0.5}/a\} \Rightarrow \{{}^{0.7}/n\},$$
$$\{{}^{0.1}/c, {}^{0.9}/a\} \Rightarrow \{a\}, \qquad \{{}^{0.1}/w\} \Rightarrow \{{}^{0.4}/c, {}^{0.4}/w\},$$
$$\{{}^{0.5}/c\} \Rightarrow \{{}^{0.7}/c\}, \qquad \{{}^{0.5}/n\} \Rightarrow \{{}^{0.5}/a, {}^{0.7}/n\},$$
$$\{{}^{0.1}/c, {}^{0.5}/a\} \Rightarrow \{{}^{0.7}/c, {}^{0.8}/a, {}^{0.7}/w\}, \{\} \Rightarrow \{{}^{0.4}/a, {}^{0.4}/n\},$$
$$\{{}^{0.1}/c, {}^{0.5}/w\} \Rightarrow \{{}^{0.7}/c, {}^{0.8}/a, {}^{0.7}/w, {}^{0.7}/n\}.$$

The basis can be seen as an additional information supplied along with the query result. Note that if Table III is considered as a classical one (no ranks, no similarities), its minimal basis consists of three (classical) FDs, namely $\{a\} \Rightarrow \{c, w, n\}$, $\{w\} \Rightarrow \{c, a, n\}$, and $\{n\} \Rightarrow \{c, a, w\}$. Thus, attributes $a$, $w$, and $n$ are all keys of the table. Contrary to the previous case with similarities and ranks, the basis does not yield any other (nontrivial) information.

## III. Future research

Future research needs to focus on further development of the relational algebra and calculi (both classical aspects like query optimization and the new ones arising due to taking degrees into account); development of functional dependencies and further types of data dependencies; data redundancy and related issues (keys, normalization in presence of similarity, preliminary results are available).

## Acknowledgment

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.

## References

[1] S. Abiteboul *et al*. The Lowell database research self-assessment. *Comm. ACM* **48**(5)(2005), 111–118.

[2] R. Belohlavek and V. Vychodil. Functional dependencies of data tables over domains with similarity relations. IICAI 2005, pages 2486–2504, Pune, India, December 2005.

[3] R. Belohlavek and V. Vychodil. 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 3882:644–658, 2006.

[4] S. Ben Yahia, H. Ounalli, and A. Jaoua. An extension of classical functional dependency: dynamic fuzzy functional dependency. *Inf. Sci*. 119:219–234, 1999.

[5] P. Bosc, D. Dubois, and H. Prade. Fuzzy functional dependencies. An overview and a critical discussion. FUZZ-IEEE'94, pages 325–330, Orlando, FL, 1994.

[6] P. Bosc, D. Kraft, and F. Petry. Fuzzy sets in database and information systems: status and opportunities. *Fuzzy Sets and Syst*. 156:418–426, 2005.

[7] B. P. Buckles and F. Petry. A fuzzy representation of data for relational databases. *Fuzzy Sets Syst*. 7:213–226, 1982.

[8] B. P. Buckles and F. E. Petry. Fuzzy databases in the new era. ACM SAC 1995, pages 497–502, Nashville, TN, 1995.

[9] B.P.Buckles, F.Petry H.Sachar. A domain calculus for fuzzy relational databases. *Fuzzy Sets Syst*. 29:327–340, 1989.

[10] J. C. Cubero and M. A. Vila. A new definition of fuzzy functional dependency in fuzzy relational datatabses. *Int. J. Intelligent Systems* 9:441–448, 1994.

[11] C. J. Date. *Database Relational Model: A Retrospective Review and Analysis*. Addison Wesley, 2000.

[12] D. Dey and S. Sarkar S. A probabilistic relational model and algebra. *ACM Trans. Dat. Syst*. 21:339–369, 1996.

[13] R. Fagin. Combining fuzzy information from multiple systems. *J. Comput. System Sci*. 58:83–99, 1999.

[14] R. Fagin. Combining fuzzy information: an overview. *ACM SIGMOD Record* 31(2):109-118, 2002.

[15] N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. Information Systems* 15:32–66, 1997.

[16] S. Gottwald. *A Treatise on Many-Valued Logics*. Research Studies Press, Beldock, England, 2001.

[17] P. Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, Dordrecht, 1998.

[18] I. F. Illyas, W. G. Aref, and A. K. Elmagarmid. Supporting top-$k$ join queries in relational databases. *The VLDB Journal* 13:207–221, 2004.

[19] R. Intan and M. Mukaidono. Fuzzy conditional probability relations and their applications in fuzzy information systems. *Knowledge and Inf. Systems* 6:345–365, 2004.

[20] G. J. Klir and B. Yuan. *Fuzzy Sets and Fuzzy Logic. Theory and Applications*. Prentice Hall, 1995.

[21] G. J. Klir. *Uncertainty and Information*. J. Wiley, 2006.

[22] C. Li, K. C.-C. Chang, I. F. Ilyas, and S. Song. RanSQL: Query Algebra and Optimization for Relational top-k queries. ACM SIGMOD 2005, pages 131–142, 2005.

[23] D. Maier. *The Theory of Relational Databases*. Computer Science Press, Rockville, 1983.

[24] H. Manilla and K.-J. Räiha. Algorithms for inferring functional dependencies from relations. *Data & Knowledge Engineering* 12:83–99, 1994.

[25] A. Natsev A, Y.-C. Chang , J. R. Smith, C.-S. Li, and J. S. Vitter. Supporting incremental join queries on ranked inputs. VLDB 2001, pages 281–290, Roma, Italy, 2001.

[26] W. Penzo. Rewriting rules to permeate complex similarity and fuzzy queries within a relational dastabase system. *IEEE Trans. Knowledge and Data Eng*. 17:255–270, 2005.

[27] H. Prade and C. Testemale. Generalizing database relational algebra for the treatment of incomplete or uncertain information and vague queries. *Inf. Sci*. 34:115–143, 1984.

[28] K. V. S. V. N. Raju, and A. K. Majumdar. Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. *ACM Trans. Database Systems* Vol. 13, No. 2:129–166, 1988.

[29] Y. Takahashi. Fuzzy database query languages and their relational completeness theorem. *IEEE Trans. Knowledge and Data Engineering* 5:122–125, February 1993.

[30] S.-L. Wang, J.-S. Tsai, and T.-P. Hong. Mining Functional Dependencies from Fuzzy Relational Databases. ACM SAC 2000, pages 490–493, Como, Italy, March 2000.