PALACKY UNIVERSITY OLOMOUC
FACULTY OF SCIENCE

Department of Informatics

The role of similarity in relational databases

Doctoral thesis
by

Lucie Ježková

2015

Abstract

In this thesis we study a particular rank-aware relational model over domains with similarities. The model results by generalization of the classical relational model of data by considering residuated lattice (instead of the two-element Boolean algebra) as the basic structure of truth values. We first focus on similarity-based functional dependencies within the model and compare them with other approaches. We also present a graph-based method of inference and show its application in closures computing. We further study the sensitivity issues and answer questions related to similarity-preservation. We show that the similarity of query results can be estimated from similarity of input data for arbitrary complex queries.

Acknowledgment

I sincerely thank my supervisor, Vilém Vychodil, for his skillful guidance.

Declaration

I hereby declare that the thesis is my original work.

This thesis is based on outcomes of the joint scientific work with
Pablo Cordero (pcordero@uma.es) and Manuel Enciso (enciso@lcc.uma.es)—chapter 3;
Vilém Vychodil (vychodil@acm.org)—chapter 4;
Radim Bělohlávek (radim.belohlavek@acm.org) and Vilém Vychodil—section 5.1 and chapter 6.

All authors contributed evenly to the results and findings contained in the respective parts.

To my beloved husband.

Contents

1 Introduction **3**

2 Preliminaries **7**

2.1 Residuated lattices 7

2.2 The relational model 11

2.3 Ranked data tables over domains with similarities 12

2.3.1 Ranked data tables 12

2.3.2 Relational operations for RDTs 13

2.3.3 Similarity-based functional dependencies 16

2.4 Directed graphs 18

3 Overview of similarity-based functional dependencies **19**

3.1 Generalizations of the relational model 20

3.2 Survey of similarity-based generalizations of FD 23

3.3 Comparison of similarity-based generalizations of FD 38

3.4 Conclusions 46

4 Derivation digraphs for graded if-then rules **49**

4.1 Derivation acyclic digraphs for FAIs 50

4.2 Completeness 53

4.3 Computing closures 57

4.4 Illustrative example 62

4.5 Conclusions 65

5 Sensitivity analysis for similarity-based functional dependencies **67**

5.1 Rank-based similarity 67

5.2 Similarity estimates for similarity-based FD 69

5.3 Conclusions 75

6 Similarity estimates of query results **77**

6.1 Similarity estimates for relational operations 77

6.1.1 Boolean-like operations 78

6.1.2 Ternary residuum 80

6.1.3 Projection and division 81
6.1.4 Similarity-based restriction 83
6.1.5 Natural and similarity-based joins 86
6.1.6 Further operations 88
6.2 Similarity of complex query results 89
6.3 Tuple-based similarity 93
6.3.1 Similarity-based semijoins and closures 93
6.3.2 Tuple-based similarity estimates 97
6.3.3 Unifying approach to similarity of RDTs 100
6.4 Conclusions 102

Summary in Czech language **103**

Chapter 1

Introduction

Relational model of data, introduced by E. F. Codd in [41], is one of the most important paradigms in Computer Science. It is based on the idea that all information should be represented by relations, which are usually depicted as tables. Roughly speaking, an n-ary relation can be seen as a table with n-columns: columns correspond to attributes and rows correspond to tuples. The set of possible values for an attribute is called a domain (or type). A key role in the design of relational databases is played by functional dependencies (FDs) [1, 7], that serve as an important tool for redundancy elimination and normalization, see [82, 53] for example. Functional dependency is a statement of the form $A \Rightarrow B$, where A, B are sets of attributes. The basic meaning of FD $A \Rightarrow B$ is that any two tuples that have the same values on all attributes from A have also the same values on all attributes from B. The meaning of "same values" is given by the identity relations, which, although not explicitly stated, are presented on each domain. These identity relations are also behind the precise matches when evaluating relational algebra queries, consider for example the `JOIN` command in SQL.

In many situations, it is desirable to consider similarities on domains instead of equalities. Assume you are looking for a hotel in Olomouc which offers a room for 100 €. Then you are interested not only in hotels offering rooms for the exact price of 100 €, but also in hotels with prices sufficiently "close" to your requirements (expressed as query in database). The "closeness" can be formalized by similarity relation, which is by nature a graded concept [118, 119]. In addition to the cases of full similarity (two prices are similar to degree 1) and full dissimilarity (two prices are similar to degree 0, i.e. not similar at all) we may say that two prices are somewhat similar (their similarity degree will be greater than 0 but smaller than 1). Functional dependencies employing similarity can reveal us new information. Let us say that we are interested whether hotels offering rooms for similar price have obtained similar average evaluation from guests (e.g. points on the scale 1-10). If not, we may conclude that there is some discrepancy between price and service quality. The classical (equality based) FD dependency: "hotels with the same room's price have the same average evaluation" will hardly be satisfied; and the violation of such FD gives us no interesting information.

As argued, the idea to use similarities for comparing domain values is quite natural and it is therefore not surprising that the concept of similarity entered the relational model very soon. The first paper on this topic was published by Buckles and Petry in 1982 [31]. Since then, several hundreds of papers dealing with similarities in relational databases have emerged, for example [92, 95, 59, 15, 44, 60]. Many papers have been devoted to functional dependencies over domains with similarities (an overview is given in Chapter 3). Such dependencies are usually trying to capture the following: "If tuples have similar values on attribute from A, then they have similar values on attributes from B. " A lot of papers also address similarity-based querying [32, 93, 59, 20], including design of similarity-based query languages [34, 74].

How to deal with the novel, similarity-based issues such as validity of similarity-based FD? In general, there are two ways: One option is to reduce the similarity-based concepts to bivalent ones: the similarity-based FD is either satisfied or not, the result of a similarity-based query is a classical (ordinary) data table, etc. A second option is to accept partial matches when evaluating queries and let the similarity-based FD to be true to some degree (between the borderline cases: satisfied, not satisfied). The second option is the one chosen by Belohlavek and Vychodil [18, 19, 20, 23, 24]. The authors built their generalization of Codd's relational model on fuzzy logic in narrow sense [67, 9, 64]. The original Codd's model have also a clear connection to a logical calculi (first order logic), which, as argued in [51], is one of the reasons that yields to its great success. Belohlavek and Vychodil extended the Codd's orignal model in the following way: domains are additionally equipped with similarity relations and each tuple in a data table has assigned a rank. The rank is a degree to which a tuple matches a similarity-based query. Both similarity degrees and ranks come from complete residuated lattice.

In this thesis we will investigate the model proposed by Belohlavek and Vychodil. In the first part of the thesis we will study similarity-based functional dependencies (SBFDs). We propose a graph-based method for reasoning and show a correspondence between construction of a directed graph and normalized proof. We also provide detailed comparison of the definition of SBFD given by Belohlavek and Vychodil with other approaches. In the second part of this thesis we examine sensitivity issues. We define two similarity measures for ranked-data tables (RDTs): ranked-based similarity (two RDTs are considered similar if they contain the same tuples with similar ranks) and tuple-based similarity (two RDTs are considered similar if they contain tuples with similar values). The tuple-based similarity can be expressed by rank-based similarity and a new relational operation, called similarity-based closure. Using the notion of rank-based similarity, we show that the similarity of query results can be estimated based on the similarity of input data prior to query execution. Such estimations can be provided for arbitrary complex queries. We further study estimations for tuple-based similarity and properties of the similarity-based closure. We also explore sensitivity issues connected to similarity-based functional dependencies.

This thesis is based on the following results:

[70] L. Ježková, P. Cordero, M. Enciso: *Codd's Relational Model of Data Over Domains With Similarities: A Comparative Survey*, Fuzzy Sets and Systems, submitted

[109] L. Urbanová and V. Vychodil: *Derivation digraphs for dependencies in ordinal and similarity-based data*, Information Sciences 268 (2014), pp. 381-–396

[12] R. Bělohlávek, L. Urbanová and V. Vychodil: *Sensitivity Analysis for Declarative Relational Query Languages with Ordinal Ranks*, In: Tompits H., Abreu S., Oetsch J., Pührer J., Seipel D., Umeda M., Wolf A. (Eds.): Applications of Declarative Programming and Knowledge Management: 19th International Conference, INAP 2011, Lecture Notes in Artificial Intelligence 7773, 2013, pp. 58–76

[11] R. Bělohlávek, L. Urbanová and V. Vychodil: *Similarity of query results in similarity-based databases*, In: Yao J. T., Ramanna S., Wang G., Suraj Z. (Eds.): Rough Sets and Knowledge Technology, Lecture Notes in Computer Science 6954, 2011, pp. 258—267

Outline of the thesis: The thesis is organized as follows.

In Chapter 2 we summarize basic facts from residuated lattices, fuzzy set theory and Codd's relational model of data. We also introduce the model proposed by Belohlavek and Vychodil. We pay attention to similarity-based relational algebra and functional dependencies.

In Chapter 3 we review and critically examine the existing work on similarity-based functional dependencies. We try to objectively compare various approaches and we propose a novel criterion to achieve this goal.

In Chapter 4 we show that degrees to which a SBFD semantically follows from sets (or graded sets) of other SBFDs can be characterized by existence of particular directed acyclic graphs with vertices labeled by attributes and degrees coming from complete residuated lattices. In addition, we show that the construction of directed acyclic graphs can be used to compute closures of sets of attributes.

In Chapter 5 we define the rank-based similarity of RDTs and show that a SBFD holds in similar data tables to similar degree. We also explore how the validity of SBFD change if we replace the antecedent (or consequent) by similar set of attributes.

In Chapter 6 we show that relational operations preserve the rank-based similarity of RDTS. We also provide an alternative definition of similarity of RDTs (tuple-based similarity) and explore its preservation for relational operations. The tuple-based similarity is closely related to a new relational operation, a similarity-based closure, whose properties are investigated as well. We also outline a general approach to similarity of RDTs that includes both the rank-based similarity and tuple-based similarity.

Chapter 2

Preliminaries

In this section we recall the basic facts of residuated lattices, fuzzy set theory, directed graphs, and relational model of data. We also introduce one extension of the Codd's model of data, namely ranked data tables over domains with similarities.

2.1 Residuated lattices

A complete residuated lattice [9, 67], which will serve as a basic structure of truth degrees, is an algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \to, 0, 1\rangle$ such that

- $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice with 0 and 1 being the least and the greatest element of L;
- $\langle L, \otimes, 1\rangle$ is a commutative monoid, i.e. $\otimes$ is a binary operation which is commutative, associative, and $a \otimes 1 = 1 \otimes a = a$ for each $a \in L$;
- $\otimes$ and $\to$ satisfy so-called adjointness property:

$$a \otimes b \leq c \text{ iff } a \leq b \to c \tag{2.1}$$

  for each $a, b, c \in L$, where $\leq$ is the order induced by the lattice structure of $\mathbf{L}$, i.e. $a \leq b$ iff $a = a \wedge b$.

Elements a of L are interpreted as truth degrees. The operations $\otimes$ and $\to$ are truth functions of "fuzzy conjunction" and "fuzzy implication" and are called a multiplication and a residuum, respectively. For every $\otimes$ there is at most one $\to$ satisfying adjointness, and similarly $\to$ uniquely determines $\otimes$. For a complete residuated lattice $\mathbf{L}$ we define

$$a \leftrightarrow b = (a \to b) \wedge (b \to a) \tag{2.2}$$

and call this derived operation a biresiduum. The biresiduum can be seen as a truth function for an equivalence, because it satisfies several properties which can be considered as natural for graded equivalence, e.g. $\leftrightarrow$ is commutative and $a \leftrightarrow b = 1$ iff $a = b$. For a nonnegative integer n, the n-th power of $a \in L$ is defined by

$$a^0 = 1 \quad \text{and} \quad a^{n+1} = a^n \otimes a. \tag{2.3}$$

The unit interval: Common examples of complete residuated lattices include structures defined on the real unit interval, i.e. structures $\mathbf{L}$ where $L = [0,1]$, $\wedge$ and $\vee$ being minimum and maximum, respectively, and $\otimes$ being a left-continuous triangular norm (shortly, a t-norm) with the corresponding $\to$. More precisely, the structure $\langle [0,1], \min, \max, \otimes, \to, 0, 1\rangle$ is a complete residuated lattice iff $\otimes$ is a left continues t-norm and $a \to b = \max\{c \,|\, a \otimes c \leq b\}$, see [67]. All complete residuated lattices on the real unit interval with continuous $\otimes$ can be constructed by means of ordinal sums [40] from the following three pairs of adjoint operations:

Łukasiewicz:	$a \otimes b = \max(a + b - 1, 0)$,	$a \to b = \min(1 - a + b, 1)$;
Gödel:	$a \otimes b = \min(a, b)$,	$a \to b = b$ if $a > b$, 1 otherwise;
Goguen:	$a \otimes b = a \cdot b$,	$a \to b = \frac{b}{a}$ if $a > b$, 1 otherwise.

Complete resitudated lattices $\langle [0,1], \min, \max, \otimes, \to, 0, 1\rangle$ with universe $[0,1]$ and with Łukasiewicz, Gödel or Goguen operations will be called standard Łukasiewicz, Gödel and Goguen algebra, respectively, and will be denoted as $[0,1]_{\text{Ł}}, [0,1]_G, [0,1]_{\Pi}$. Sometimes we will denote $\to_{\text{Ł}}$, $\to_G$, and $\to_{\Pi}$ to emphasize the Łukasiewicz, Gödel and Goguen implication, respectively.

Theorem 1 ([9])**.** *Each residuated lattice satisfies:*

$$a \leq b \quad \textit{iff} \quad a \to b = 1, \tag{2.4}$$

$$b_1 \leq b_2 \quad \textit{implies} \quad a \to b_1 \leq a \to b_2, \tag{2.5}$$

$$a_1 \leq a_2 \quad \textit{implies} \quad a_2 \to b \leq a_1 \to b, \tag{2.6}$$

$$a \to 1 = 1, \tag{2.7}$$

$$0 \to a = 1, \tag{2.8}$$

$$1 \to a = a, \tag{2.9}$$

$$a \otimes b \leq a, \tag{2.10}$$

$$a \leq b \to a, \tag{2.11}$$

$$a \otimes b \leq a \wedge b, \tag{2.12}$$

$$a \otimes (a \to b) \leq b, \tag{2.13}$$

$$(a \otimes b) \to c = a \to (b \to c), \tag{2.14}$$

$$a \otimes (b \to c) \leq b \to (a \otimes c), \tag{2.15}$$

$$(a \to b) \otimes (b \to c) \leq a \to c, \tag{2.16}$$

$$a \otimes \bigvee_{i \in I} b_i = \bigvee_{i \in I} (a \otimes b_i), \tag{2.17}$$

$$a \to \bigwedge_{i \in I} b_i = \bigwedge_{i \in I} (a \to b_i), \tag{2.18}$$

$$\bigvee_{i \in I} a_i \to b = \bigwedge_{i \in I} (a_i \to b), \tag{2.19}$$

$$\bigwedge_{i\in I}(a_i \to b_i) \leq \bigwedge_{i\in I} a_i \to \bigwedge_{i\in I} b_i, \tag{2.20}$$

$$a \otimes \bigwedge_{i\in I} b_i \leq \bigwedge_{i\in I}(a \otimes b_i), \tag{2.21}$$

$$\bigwedge_{i\in I} a_i \otimes \bigwedge_{i\in I} b_i \leq \bigwedge_{i\in I}(a_i \otimes b_i). \tag{2.22}$$

We now turn our attention to unary operations called truth-stressing hedges [68, 67, 58]. Let $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \to, 0, 1\rangle$ be a complete residuated lattice. An unary operation $^* : L \to L$ satisfying

$$1^* = 1, \tag{2.23}$$

$$a^* \leq a, \tag{2.24}$$

$$(a \to b)^* \leq a^* \to b^*, \tag{2.25}$$

$$a^{**} = a^*, \tag{2.26}$$

for each $a, b \in L$ will be called a truth-stressing hedge (or shortly hedge) for $\mathbf{L}$. The algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \to, ^*, 0, 1\rangle$ is then called a complete residuated lattice with hedge and denoted as $\mathbf{L^*}$. Hedge * can be understood as a truth function of unary connective "very true". If ϕ is a proposition with truth degree $||\phi||$, then the truth degree of proposition "ϕ is very true" (or "it is very true that ϕ") is $||\phi||^*$. Properties (2.23)–(2.26) have natural interpretations, e.g., (2.24) can be read: "if a is very true, then a is true", (2.25) can be read: "if $a \to b$ is very true and if a is very true, then b is very true", etc.

Two boundary cases of (truth-stressing) hedges are

(i) identity, i.e., $a^* = a$ $(a \in L)$;

(ii) globalization [105]:

$$a^* = \begin{cases} 1, & \text{if } a = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.27}$$

If * is a globalization, then $(a \to b)^* = 1$ iff $a \to b = 1$ iff $a \leq b$.

Since * is intensive (2.24), monotone (consequence of (2.23) and (2.25)) and idempotent (2.26), it is an interior operator. We may therefore denote by $\mathrm{fix}(*)$ the set of all fixed points:

$$\mathrm{fix}(*) = \{a \in L \,|\, a^* = a\} = \{a^* \,|\, a \in L\}. \tag{2.28}$$

If $*_1$ and $*_2$ are two hedges on $\mathbf{L}$ such that $\mathrm{fix}(*_1) \subseteq \mathrm{fix}(*_2)$ we say that $*_1$ is stronger than $*_2$.

A special case of a complete residuated lattice with hedge is the two-element Boolean algebra $\langle\{0, 1\}, \wedge, \vee, \otimes, \to, ^*, 0, 1\rangle$, denoted by $\mathbf{2}$, which is the structure of truth degrees of the classical logic. That is, the operations $\wedge, \vee, \otimes, \to$ of $\mathbf{2}$ are the truth functions of the corresponding logical connectives of the classical logic and $0^* = 0$, $1^* = 1$.

L-sets and L-relations [9, 118]

An **L**-set (or fuzzy set) in a universe X is a mapping $A : X \to L$, where L is a support of a complete residuated lattice **L**. The degree $A(x)$ is interpreted as a degree to which an element x belongs to A. The set of all **L**-sets in X is denoted by L^X. We are going to use the following notation for denoting **L**-sets: If $X = \{x_1, \ldots, x_n\}$ then an **L**-set A in X can be denoted by $A = \{{}^{a_1}/_{x_1}, \ldots, {}^{a_n}/_{x_n}\}$ meaning that $A(x_i)$ equals a_i for each $i = 1, \ldots, n$. Operations with **L**-sets are defined component-wise, for $A, B \in L^X$ we have:

$$(A \cup B)(u) = A(u) \vee B(u), \tag{2.29}$$

$$(A \cap B)(u) = A(u) \wedge B(u), \tag{2.30}$$

$$(A \otimes B)(u) = A(u) \otimes B(u), \tag{2.31}$$

$$(A \to B)(u) = A(u) \to B(u). \tag{2.32}$$

For **L**-sets $A, B \in L^X$ we define a degree of subsethood of A in B and a degree of equality of A, B as follows:

$$S(A, B) = \bigwedge_{x \in X} (A(x) \to B(x)), \tag{2.33}$$

$$E(A, B) = \bigwedge_{x \in X} (A(x) \leftrightarrow B(x)). \tag{2.34}$$

The subsethood relation (2.33) generalizes the classical subsethood relation "$\subseteq$". In particular, we have $S(A, B) = 1$ (A is fully included in B) iff $A(x) \leq B(x)$ for each $x \in X$. $S(A, B)$ can be understood as a truth degree of the following formula: "for every $x \in X$: if x belongs to A, then x belongs to B." And similarly $E(A, B)$ can be thought of as a truth degree of the formula "for every $x \in X$: x belongs to A iff x belongs to B." The following Theorem shows some properties of graded subsethood S and graded equality E.

Theorem 2 ([9]). *For* **L***-sets* $A, B, C \in L^X$ *we have:*

$$S(A, A) = 1, \tag{2.35}$$

$$S(A, B) \otimes S(B, C) \leq S(A, C), \tag{2.36}$$

$$E(A, A) = 1, \tag{2.37}$$

$$E(A, B) = E(B, A), \tag{2.38}$$

$$E(A, B) \otimes E(B, C) \leq E(A, C). \tag{2.39}$$

An n-ary **L**-relation between sets $X_1, \ldots, X_n$ is an **L**-set $I \in L^{X_1 \times \ldots \times X_n}$. Thus a binary **L**-relation on X is a mapping $I : X \times X \to L$ that assigns to each pair of elements $x, y \in X$ a degree to which they are related according to I. A binary **L**-relation I on X is called an **L**-equivalence if it is reflexive, symmetric and $\otimes$-transitive, that is for all $x, y, z \in X$:

$$I(x, x) = 1, \tag{2.40}$$

$$I(x, y) = I(y, x), \tag{2.41}$$

$$I(x,y) \otimes I(y,z) \leq I(x,z). \tag{2.42}$$

L-equivalence, or fuzzy equivalence, will be denoted as $\equiv$. Note that (2.34) is an **L**-equivalence on L^X. A binary **L**-relation I that is reflexive and symmetric will be called similarity and denoted as $\approx$. We will write $x \approx y$ instead of $\approx (x,y)$. An **L**-equivalence that satisfies separability

$$I(x,y) = 1 \quad \text{iff} \quad x = y$$

will be called **L**-equality. It is important to keep in mind that $I(x,y)$ are general degrees from L and thus have a comparative meaning. $I(x_1,y_1) > I(x_2,y_2)$ means that the values x_1 and y_1 are more related according to I than the values of x_2 and y_2. Moreover, since 1 is the greatest element of **L**, $I(x,y) = 1$ means that x and y are fully related according to I. For 0 being the least element of **L**, the meaning of $I(x,y) = 0$ is that x and y are not related (according to I) at all.

2.2 The relational model

Now we will present the basic notions from the relational model of data, which was introduced by Codd in [41]. For further details see [82, 52]. Let Y denotes a set of attribute names. For each attribute $y \in Y$ we consider its domain D_y, which is an arbitrary nonempty set of all values allowed for y. A relation scheme is a finite subset $R \subseteq Y$. In particular $R = \emptyset \subseteq Y$ is an empty relation scheme. For each relation scheme R, $\mathrm{Tupl}(R)$ denotes $\prod_{y \in R} D_y$, i.e. the Cartesian product of domains D_y ($y \in R$). Recall that the Cartesian product is a set of all maps $r\colon R \to \bigcup_{y \in R} D_y$ such that $r(y) \in D_y$ holds for all $y \in R$. For $R = \emptyset$ we get $\prod_{y \in \emptyset} D_y = \{\emptyset\}$. A data table $\mathcal{D}$ on R is any finite subset of $\mathrm{Tupl}(R)$. Each $r \in \mathrm{Tupl}(R)$ is called a *tuple over* R and $r(y)$ is called the *y-value of* r. The only data tables on relation scheme $R = \emptyset$ are $\mathcal{D}_\top = \{\emptyset\}$ and $\mathcal{D}_\bot = \emptyset$ which are called `TABLE_DEE` and `TABLE_DUM` (in [54]) and represent the truth values 1 and 0, respectively. Moreover, for each $A \subseteq R$, the restriction of r to the subset A is denoted by $r(A)$, that is $r(A)\colon A \to \bigcup_{y \in A} D_y$. If $\mathcal{D}$ is a data table on a relation scheme R, i.e. $\mathcal{D} \subseteq \mathrm{Tupl}(R)$, and A is a subset of R, then $\pi_A(\mathcal{D})$ denotes the projection of the data table $\mathcal{D}$ to the set of attributes A,

$$\pi_A(\mathcal{D}) = \{r(A) \mid r \in \mathcal{D}\}. \tag{2.43}$$

Assume A, B are sets of attributes, i.e. $A, B \subseteq R$, then we say A determines B (or B is functionally dependent on A) if whenever two tuples of $\mathcal{D}$ agree on attributes from A then they agree on attributes from B. We write $A \Rightarrow B$ and call such a statement functional dependency (FD). Formally, FD is satisfied by relation $\mathcal{D}$ iff

$$\forall r_1, r_2 \in \mathcal{D}: \text{ if } r_1(A) = r_2(A), \text{ then } r_1(B) = r_2(B). \tag{2.44}$$

We will denote by $||A \Rightarrow B||_{\mathcal{D}}$ the degree to which an FD $A \Rightarrow B$ holds in a relation $\mathcal{D}$. From (2.44) we obviously have $||A \Rightarrow B||_{\mathcal{D}} \in \{0,1\}$.

2.3 Ranked data tables over domains with similarities

There are various extensions of the Codd's relational model of data which take similarities into account. Their comparison, mainly focused on similarity based functional dependencies, is presented in Section 3. Here we concentrate on the approach originally introduced by Belohlavek and Vychodil, see [15, 18, 23, 24].

2.3.1 Ranked data tables

The concept of a ranked data table over domains with similarities is the counterpart to the concept of a relation on a relation scheme. As in the original Codd's relational model, Y denotes a set of attributes names, a relation scheme is any finite subset $R \subseteq Y$, and a domain D_y is a set of all possible values of the attribute $y \in Y$. The relational model is generalized in the following way:

(i) Each domain D_y is additionally equipped with a similarity relation $\approx_y$, i.e. with a reflexive symmetric binary **L**-relation on D_y;

(ii) Each tuple has assigned a rank, which represents a degree to which a tuple matches a query. Ranks have mainly comparative meaning: the higher the rank the better the match.

Similarity degrees as well as ranks come from complete residuated lattice. The following table which can be seen as a result of the query "hotels in Olomouc with a room for 100 €" is an example of a ranked data table.

	name	*price*	*eval*	*dist*
1.00	Hotel Central	100,00 €	8.9	0.5 km
0.90	Hotel ABC	90,00 €	9.1	0.8 km
0.85	Pension Angel	115,00 €	8.5	1.2 km
0.45	Hotel Paradise	55,00 €	6.7	2.5 km
0.30	Hotel Kryton	170,00 €	10.0	1.6 km

In the data table we store the following informations: ***name*** (name of the hotel), ***price*** (price for the double room), ***eval*** (average evaluation), ***dist*** (distance from the city center). The numbers $1.00, \ldots, 0.30$ in the leftmost column are the ranks from a scale of truth values (here $[0, 1]$). The remaining part of the table can be seen as a classical data table. Similarities on domains are not shown directly. For the attribute ***price***, we consider the following similarity on its domain: $p_1 \approx_{price} p_2 = (100 - |p_1 - p_2|)/100$ if $|p_1 - p_2| < 100$, and 0 otherwise.

We will now introduce ranked data tables (RDTs) formally:

Definition 3 ([23]). *Let $R \subseteq Y$ be a relation scheme and let $\langle D_y, \approx_y, \rangle$ be domains with similarities for attributes $y \in R$. A ranked data table on R over $\{\langle D_y, \approx_y, \rangle \,|\, y \in R\}$ is any map*

$$\mathcal{D} : \prod_{y \in R} D_y \to L, \tag{2.45}$$

such that the set $\{r \in \prod_{y \in R} D_y \mid \mathcal{D}(r) > 0\}$*, called the answer set, is finite. The cardinality of the answer set of* $\mathcal{D}$ *is called the* size *of* $\mathcal{D}$ *and is denoted by* $|\mathcal{D}|$*.* $\mathcal{D}$ *is called* nonranked *if* $\mathcal{D}(r) \in \{0, 1\}$ *for any* r*. Each degree* $\mathcal{D}(r) \in L$ *is called a* rank *of* r *in* $\mathcal{D}$*.*

Remark 1. *The cardinality of the answer set is influenced by the choice of residuated lattice* $\mathbf{L}$*. It was shown in [110] that ordinal sums play an important role in altering the size of the answer set.*

Each RDT $\mathcal{D}$ on the empty relation scheme is uniquely given by the degree to which the empty tuple belongs to $\mathcal{D}$, i.e. by $\mathcal{D}(\emptyset)$.

Definition 4 ([23])**.** *For each* $a \in L$*, we denote by* $a_\emptyset$ *the RDT on* $\emptyset$ *such that* $a_\emptyset(\emptyset) = a$*.*

Therefore, each $a_\emptyset$ is a map which assigns to the empty tuple the degree $a \in L$

$$a_\emptyset \colon \{\emptyset\} \to L. \tag{2.46}$$

Each $a_\emptyset$ is viewed as a relational representation of the rank $a \in L$. Notice the analogy with the classical model: the role of `TABLE_DEE` and `TABLE_DUM` is now played by RDTs $1_\emptyset$ and $0_\emptyset$ which are both particular cases of (2.46).

Note that the original Codd's model is a particular case of the model of RDT over domains with similarities. If one takes the two-element Boolean algebra for $\mathbf{L}$, then all RDTs become nonranked and all similarities become identities.

2.3.2 Relational operations for RDTs

We introduce relational operations for RDTs as given in [23]. For RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$ on relation scheme R, we put

$$(\mathcal{D}_1 \cup \mathcal{D}_2)(r) = \mathcal{D}_1(r) \vee \mathcal{D}_2(r), \tag{2.47}$$

$$(\mathcal{D}_1 \cap \mathcal{D}_2)(r) = \mathcal{D}_1(r) \wedge \mathcal{D}_2(r), \tag{2.48}$$

$$(\mathcal{D}_1 \otimes \mathcal{D}_2)(r) = \mathcal{D}_1(r) \otimes \mathcal{D}_2(r). \tag{2.49}$$

$\mathcal{D}_1 \cup \mathcal{D}_2$ is called the *union* of $\mathcal{D}_1$ and $\mathcal{D}_2$. $\mathcal{D}_1 \cap \mathcal{D}_2$ and $\mathcal{D}_1 \otimes \mathcal{D}_2$ are called the $\wedge$*-intersection* and the $\otimes$*-intersection*. Hence, $\cup$, $\cap$, and $\otimes$ are defined componentwise based on the operations of the complete residuated lattice $\mathbf{L}$ and can be seen as counterparts to the ordinary set-theoretic operations with relations on relation schemes. The fact that we have two kinds of intersection for RDTs is natural, since both $\wedge$ and $\otimes$ are fundamental operations of residuated lattices that generalize the classical conjunction and are not mutually definable in general. The intersection based on $\wedge$ is also called an *idempotent intersection* and the intersection based on $\otimes$ is called a *strong intersection* and it is not idempotent in general, i.e. for $\mathcal{D}$ on R, we may have $\mathcal{D}(r) \otimes \mathcal{D}(r) < \mathcal{D}(r)$. As we have mentioned, the meaning of a rank is a degree to which a tuple matches a query. Therefore if $\mathcal{D}_1$ is viewed as a result of query Q_1 and $\mathcal{D}_2$ is viewed as a result of query Q_2, then the rank $(\mathcal{D}_1 \cup \mathcal{D}_2)(r)$ is "a degree to which r matches Q_1 or r matches Q_2".

Note that a data table resulting from componentwise application of $\to$ may have in general (if at least one of the domain is infinite) an infinite number of tuples with nonzero ranks. This is a consequence of (2.8). In order to have a domain independent reisuduum, the authors introduced a ternary counterpart of $\to$ with one of the argument serving as a range [23]. For RDTs $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ on relational scheme R we put

$$\big(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2\big)(r) = \mathcal{D}_3(r) \otimes \big(\mathcal{D}_1(r) \to \mathcal{D}_2(r)\big) \tag{2.50}$$

for all $r \in \mathrm{Tupl}(R)$. $\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2$ is called a *residuum* of $\mathcal{D}_1$ with respect to $\mathcal{D}_2$ which ranges over $\mathcal{D}_3$. From Equation (2.10) it immediately follows that $\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2 \subseteq \mathcal{D}_3$. The RDT $\mathcal{D}_3$ serves as a range for the componentwise application of residuum $\to$, which is more easily seen if one considers $\mathcal{D}_3$ as a nonranked RDT. In this case $\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2$ can be rewritten as follows:

$$\big(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2\big)(r) = \begin{cases} \mathcal{D}_1(r) \to \mathcal{D}_2(r), & \text{if } \mathcal{D}(r) = 1, \\ 0, & \text{otherwise.} \end{cases}$$

If the RDTs $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ are considered as results of queries Q_1, Q_2, Q_3, respectively, then $\big(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2\big)(r)$ is a degree to which "r matches Q_3 and if it matches Q_1 then it matches Q_2." If we take the first or the second argument of the ternary residuum as a constant degree from L, we obtain two important binary operations: residuated c-negation and residuated c-shift. For RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$ on R and for $c \in L$, we put

$$\big(\mathcal{D}_2 \boxbslash_c \mathcal{D}_1\big)(r) = \mathcal{D}_1(r) \twoheadrightarrow^{\mathcal{D}_2(r)} c, \tag{2.51}$$

$$\big(c \twoheadrightarrow^{\mathcal{D}_2} \mathcal{D}_1\big)(r) = c \twoheadrightarrow^{\mathcal{D}_2(r)} \mathcal{D}_1(r), \tag{2.52}$$

for all tuples $r \in \mathrm{Tupl}(R)$. The operation defined by (2.51) is called a *residuated c-negation* of $\mathcal{D}_1$ which ranges over $\mathcal{D}_2$. In a particular case if $c = 0$, we can abbreviate $\mathcal{D}_2 \boxbslash_c \mathcal{D}_1$ by $\mathcal{D}_2 \boxbslash \mathcal{D}_1$ and call it a (residuated) negation of $\mathcal{D}_1$ which ranges over $\mathcal{D}_2$. Moreover, (2.52) is called a *residuated c-shift* of $\mathcal{D}_1$ which ranges over $\mathcal{D}_2$. If the RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$ are results of queries Q_1 and Q_2, then $\big(\mathcal{D}_2 \boxbslash_c \mathcal{D}_1\big)(r)$ is a degree to which "r matches Q_2 and r matches Q_1 at most to degree c." Note that if $c = 0$, then the meaning of $\big(\mathcal{D}_2 \boxbslash_c \mathcal{D}_1\big)(r)$ is a degree to which "r matches Q_2 and r does not match Q_1." Similarly $\big(c \twoheadrightarrow^{\mathcal{D}_2} \mathcal{D}_1\big)(r)$ is a degree to which "r matches Q_2 and r matches Q_1 at least to degree c."

Projections and residuated divisions represent operations which allow users to express queries with existential and universal quantification.

We start by considering the projection. If $\mathcal{D}$ is an RDT on R_1, the *projection* of $\mathcal{D}$ onto $R_2 \subseteq R_1$, denoted by $\pi_{R_2}(\mathcal{D})$, is defined as

$$(\pi_{R_2}(\mathcal{D}))(r_2) = \bigvee_{r_3 \in \mathrm{Tupl}(R_1 \setminus R_2)} \mathcal{D}(r_2 r_3) \tag{2.53}$$

for each $r_2 \in \mathrm{Tupl}(R_2)$. Note that (2.53) uses a general suprema $\bigvee$ to define the rank of r_2 in $\pi_{R_2}(\mathcal{D})$. If $\mathcal{D}$ is interpreted as a result of query Q, then the rank of r_2 in $\pi_{R_2}(\mathcal{D})$ can be understood as the degree to which "there is a tuple matching Q which agrees with r_2 on all the attributes from R_2".

Relational expressions involving projections can be utilized in existentially quantified queries. In the same spirit, relational expressions involving divisions are algebraic counterpart to universally quantified queries, see [82]. Since in residuated logics the existential and universal quantifiers are not mutually definable [64, 67], the residuated division is introduced as a fundamental operation. Moreover, the residuated division is considered as a ternary operation in order to ensure its domain independence.

Let $\mathcal{D}_1$ be an RDT on R_1, $\mathcal{D}_2$ be an RDT on $R_2 \subseteq R_1$, and $\mathcal{D}_3$ be an RDT on $R_3 = R_1 \setminus R_2$. Then, a *division* of $\mathcal{D}_1$ by $\mathcal{D}_2$ which ranges over $\mathcal{D}_3$ is an RDT on R_3 denoted by $\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2$ and defined by

$$\begin{aligned}\big(\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2\big)(r_3) &= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_2(r_2) \twoheadrightarrow^{\mathcal{D}_3(r_3)} \mathcal{D}_1(r_2 r_3)\big) \\ &= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_3(r_3) \otimes (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2 r_3))\big) \end{aligned} \tag{2.54}$$

for each $r_3 \in \mathrm{Tupl}(R_3)$. It is easily seen that $\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2 \subseteq \mathcal{D}_3$ and therefore the result of division is fully contained in $\mathcal{D}_3$. Therefore $\mathcal{D}_3$ can be seen as a range for the division.

Similarity-based restriction is another fundamental operation and it is a counterpart to the ordinary restriction. If $\mathcal{D}$ is an RDT on relation scheme R, $y \in R$ and $d \in D_y$, the *similarity-based restriction* of $\mathcal{D}$ by $y \approx d$ is an RDT on R denoted by $\sigma_{y \approx d}(\mathcal{D})$ and defined by

$$(\sigma_{y \approx d}(\mathcal{D}))(r) = \mathcal{D}(r) \otimes r(y) \approx_y d, \tag{2.55}$$

for all $r \in \mathrm{Tupl}(R)$. The meaning of (2.55) is best seen if $\mathcal{D}$ is viewed as a result of query Q. Then, the rank of r given by (2.55) is a degree to which "r matches Q *and* $r(y)$ is similar to d", where the logical connective *and* is interpreted by $\otimes$. The similarity-based restriction that compares values of two attributes y_1, y_2 with the same domain is introduced as follows: For an RDT $\mathcal{D}$ on relation scheme R and for $y_1, y_2 \in R$ such that $D_{y_1} = D_{y_2}$ and $u \approx_{y_1} v = u \approx_{y_2} v$ for all $u, v \in D_{y_1}$, we define

$$(\sigma_{y_1 \approx y_2}(\mathcal{D}))(r) = \mathcal{D}(r) \otimes r(y_1) \approx_{y_1} r(y_2). \tag{2.56}$$

By applying a similarity-based restriction to a nonranked RDT we obtain a ranked RDT.

The (equality-based) natural join is introduced as follows. If $\mathcal{D}_1$ is an RDT on relation scheme $R_1 \cup R_3$ and $\mathcal{D}_2$ is an RDT of relation scheme $R_2 \cup R_3$ such that $R_1 \cap R_2 = R_1 \cap R_3 = R_2 \cap R_3 = \emptyset$ (i.e., R_1, R_2, and R_3 are pairwise disjoint), then the (*equality-based*) *natural join* of $\mathcal{D}_1$ and $\mathcal{D}_2$ is an RDT on relation scheme $R_1 \cup R_2 \cup R_3$ denoted by $\mathcal{D}_1 \bowtie \mathcal{D}_2$ and defined by

$$\big(\mathcal{D}_1 \bowtie \mathcal{D}_2\big)(r_1 r_2 r_3) = \mathcal{D}_1(r_1 r_3) \otimes \mathcal{D}_2(r_2 r_3) \tag{2.57}$$

for each $r_1 \in \mathrm{Tupl}(R_1)$, $r_2 \in \mathrm{Tupl}(R_2)$, and $r_3 \in \mathrm{Tupl}(R_3)$. If $\mathcal{D}_1$ is a result of query Q_1 and $\mathcal{D}_2$ is a result of query Q_2, then the rank $\big(\mathcal{D}_1 \bowtie \mathcal{D}_2\big)(r_1 r_2 r_3)$ is a degree to which "$r_1 r_2$ matches Q_1 and $r_2 r_3$ matches Q_2. Natural joins have important special cases:

i) Considering $\mathcal{D}_1$ and $\mathcal{D}_2$ in (2.57) with $R_3 = \emptyset$, we get a natural join $\mathcal{D}_1 \bowtie \mathcal{D}_2$ of two RDTs on disjoint relation schemes. This generalizes the traditional *cross join*.

ii) Considering $R_1 = R_2 = \emptyset$, then $\mathcal{D}_1 \bowtie \mathcal{D}_2$ is a $\otimes$-intersection.

Similarity-based restrictions can be used to define various types of similarity-based joins. The first type of join we introduce is a similarity-based equijoin. For RDTs $\mathcal{D}_1$ on R_1 and $\mathcal{D}_2$ on R_2 such that $R_1 \cap R_2 = \emptyset$, the *similarity-based equijoin* of $\mathcal{D}_1$ and $\mathcal{D}_2$ by $y_1 \approx y_2$, denoted by $\mathcal{D}_1 \bowtie_{y_1 \approx y_2} \mathcal{D}_2$, is defined by

$$\mathcal{D}_1 \bowtie_{y_1 \approx y_2} \mathcal{D}_2 = \sigma_{y_1 \approx y_2}(\mathcal{D}_1 \bowtie \mathcal{D}_2), \tag{2.58}$$

provided that $y_1 \in R_1$, $y_2 \in R_2$, and both y_1 and y_2 have the same domain with similarity. A second type of join can be used when we want to put only a partial emphasis instead of the full emphasis on the similarity-based condition $y_1 \approx y_2$:

$$(\mathcal{D}_1 \bowtie_{c/_{y_1 \approx y_2}} \mathcal{D}_2)(r_1 r_2) = \mathcal{D}_1(r_1) \otimes \mathcal{D}_2(r_2) \otimes (c \to r_1(y_1) \approx_{y_1} r_2(y_2)) \tag{2.59}$$

for any $r_1 \in \mathrm{Tupl}(R_1)$ and $r_2 \in \mathrm{Tupl}(R_2)$. As a result of property (2.4) the degree $c \to r_1(y_1) \approx_{y_1} r_2(y_2)$ is indeed "a degree to which $r_1(y_1)$ is similar to $r_2(y_2)$ at least to degree $c \in L$." Note that for $c = 1$, (2.59) becomes (2.58), which is a consequence of (2.9).

We have seen that similarity-based restriction can produce a ranked RDT from a non-ranked one. Conversely, operations kernel and support produce a nonranked RDT from a data table containing ranks. For $\mathcal{D}$, we define RDTs $\Delta\mathcal{D}$ (a *kernel* of $\mathcal{D}$) and $\nabla\mathcal{D}$ (a *support* of $\mathcal{D}$) on the same relation scheme as follows:

$$(\Delta\mathcal{D})(r) = \begin{cases} 1, & \text{if } \mathcal{D}(r) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2.60}$$

$$(\nabla\mathcal{D})(r) = \begin{cases} 1, & \text{if } \mathcal{D}(r) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{2.61}$$

If $\mathcal{D}$ is interpreted as a result of query Q, the kernel $\Delta\mathcal{D}$ of $\mathcal{D}$ contains tuples which match the query Q fully (to a degree 1), whereas $\nabla\mathcal{D}$ is an RDT which consists of all tuples that match Q to a nonzero degree.

The last operation we discuss is renaming, which plays the same role as in the Codd's model. Given an RDT $\mathcal{D}$, the *renaming* $\rho_f(\mathcal{D})$ produces an RDT with the same contents (and the same ranks) with attributes renamed by an injective renaming function $f : R \to Y$ such that attributes y and $f(y)$ have the same domain.

The authors proved in [20, 23] that relational algebra has the same expressive power as the domain relational calculus (with range declarations). The domain relational calculus is based on first-order fuzzy logic.

2.3.3 Similarity-based functional dependencies

In this section we introduce functional dependencies, which are called similarity-based functional dependencies (SBFDs) and their interpretation in RDTs [15, 18, 21, 24]. For

$A, B \in L^R$ the similarity-based functional dependency is an expression of the form $A \Rightarrow B$. For an RDT $\mathcal{D}$ on R a degree $||A \Rightarrow B||_{\mathcal{D}}$ to which $A \Rightarrow B$ is true in $\mathcal{D}$ is defined by

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} \Big((r_1(A) \approx r_2(A))^* \rightarrow (r_1(B) \approx r_2(B)) \Big), \tag{2.62}$$

where

$$r_1(A) \approx_{\mathcal{D}} r_2(A) = (\mathcal{D}(r_1) \otimes \mathcal{D}(r_2)) \rightarrow \bigwedge_{y \in R} (A(y) \rightarrow r_1(y) \approx_y r_2(y)). \tag{2.63}$$

The authors built their approach on first-order predicate fuzzy logic [67] and thus (2.62) is the truth degree of the following formula: "For all pairs of tuples: if r_1 and r_2 have very similar values on attributes from A then r_1 and r_2 have similar values on attributes from B." And (2.63) is the truth degree of the formula: "If r_1, r_2 are from $\mathcal{D}$ then for each attribute y from A, r_1 and r_2 have similar values on y."

Remark 2. *(i) If A is a crisp set and all similarities become identities then $r_1(A) \approx_{\mathcal{D}} r_2(A) = 1$ iff r_1 and r_2 are equal on all attributes from A. (ii) The antecedent of* (2.62) *is modified by a hedge. The hedge can be seen as an additional parameter which influences the truth degree of $A \Rightarrow B$.*

In what follows, we are interested in the entailment of SBFDs from theories [15, 24]. An **L**-set T of SBFDs on R will be called a *theory.* A theory T is called *crisp* if $T(A \Rightarrow B) \in \{0, 1\}$ for each SBFD $A \Rightarrow B$. We say that an RDT $\mathcal{D}$ is a model of theory T whenever $T(A \Rightarrow B) \leq ||A \Rightarrow B||_{\mathcal{D}}$ for all $A \Rightarrow B$ on R. Put into words: an RDT $\mathcal{D}$ is a model of theory T if for all $A \Rightarrow B$ from T the degree to which $A \Rightarrow B$ is true in $\mathcal{D}$ is at least as high as a degree to which $A \Rightarrow B$ belongs to T (is prescribed by T). The collection of models will be denoted as $\mathrm{Mod}(T)$, i.e.

$$\mathrm{Mod}(T) = \{\mathcal{D} \,|\, \text{for each } A, B \in L^R : T(A \Rightarrow B) \leq ||A \Rightarrow B||_{\mathcal{D}}\}, \tag{2.64}$$

where $\mathcal{D}$ is any RDT over R. A degree $||A \Rightarrow B||_T$ to which $A \Rightarrow B$ (on R) *semantically follows* from T is defined by

$$||A \Rightarrow B||_T = \textstyle\bigwedge_{\mathcal{D} \in \mathrm{Mod}(T)} ||A \Rightarrow B||_{\mathcal{D}}. \tag{2.65}$$

The degree to which a particular SBFD follows from a given theory (an **L**-set of SBFDs) can be expressed using the concepts of entailment to degree 1 and crisp theory. More precisely: For $A, B \in L^R$ and theory T on R

$$||A \Rightarrow B||_T = \bigvee \{c \in L \,|\, ||A \Rightarrow c \otimes B||_{\mathrm{crisp}(T)} = 1\}, \tag{2.66}$$

where $\mathrm{crisp}(T) = \{A \Rightarrow T(A \Rightarrow B) \otimes B \,|\, A, B \in L^R \text{ and } T(A \Rightarrow B) \otimes B \not\subseteq A\}$.

We have introduced the concept of semantic entailment, which is defined in terms of models. As in the ordinary Codd model there is an alternative type of entailment based on the notion of provability. The deductive system for SBFDs consists of three rules:

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

(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 $A, B, C, D \in L^R$ and $c \in L$. The inference system consisting of (Ax), (Cut), and (Mul) is complete in the following sense: $||A \Rightarrow B||_T = 1$ iff $T \vdash A \Rightarrow B$, i.e., iff there is a proof of $A \Rightarrow B$ from T. The proof of $A \Rightarrow B$ from T is a sequence of SBFDs ending with $A \Rightarrow B$ such that each element of the sequence is either from T or is inferred from the preceding formulas using (Ax), (Mul), or (Cut). This result (ordinary-style completeness) characterizes SBFDs which follow semantically from T to degree 1. There is also a result on graded-style (Pavelka-style [89, 90, 91]) completeness saying that

$$||A \Rightarrow B||_T = \bigvee \{c \in L \,|\, T \vdash A \Rightarrow c \otimes B\}, \tag{2.67}$$

i.e., the degree to which $A \Rightarrow B$ semantically follows from T is a supremum of degrees $c \in L$ for which $A \Rightarrow c \otimes B$ is provable from T in the ordinary sense. The completeness results have been established over all finite residuated lattices and general complete residuated lattices (considering an additional infinitary deduction rule), see [14].

2.4 Directed graphs

We now recall basic notions of directed graphs [5].
A directed graph (a digraph) is a pair $\mathbf{D} = \langle V, A \rangle$, where V is a nonempty finite set of elements called vertices and A is a binary relation $A \subseteq V \times V$, each $\langle v, w \rangle \in A$ is called an arc. V and A are called the vertex set and the arc set of $\mathbf{D}$, respectively. If $\langle v, w \rangle \in A$, we say that the arc $\langle v, w \rangle$ leaves v and enters w. A digraph $\mathbf{D} = \langle V, A \rangle$ is acyclic (in short, $\mathbf{D}$ is a DAG) if there is no finite sequence $v_1, \ldots, v_k$ ($k \geq 2$) of vertices from V such that $v_1 = v_k$ and $\langle v_i, v_{i+1} \rangle \in A$ for all $i = 1, \ldots, k-1$, i.e., if $\mathbf{D}$ does not contain a cycle in the usual sense.

Chapter 3

Overview of similarity-based functional dependencies

We have seen one particular extension of functional dependencies, namely similarity-based functional dependencies proposed by Belohlavek and Vychodil, see Section 2.3.3. This approach is one of many extensions that appeared in the past, actually more than one hundred papers dealing with functional dependencies (FDs) over domain with similarities can be found in the literature. In our opinion the wide variety of approaches are worthy of an exhaustive review and comparison.

The name "fuzzy functional dependencies" is often used for various extensions of FDs which we think is unfortunate for several reasons. First of all, the term "fuzzy functional dependency" is usually used for functional dependencies defined within "fuzzy relational model". But there is no agreement among researchers what the terms "fuzzy relational model" or "fuzzy database" really mean. For example in [107] the term "fuzzy database" is used for a collection of (ordinary, crisp) relations defined over complex domains (sets of possibility distributions). Contrary to that, in [94] the term "fuzzy relational data model" is used for collection of fuzzy relations, i.e. fuzzy subsets of the Cartesian product of domains. Moreover, although many definitions of so called fuzzy functional dependencies extend the classical one, the dependency usually remains crisp in the sense that either a given relation satisfies the dependency or it does not. In this sense the term fuzzy functional dependency is somehow inadequate. We will therefore use the term generalized functional dependency (GFD) and generalized relational model (GRM) to prevent misunderstanding.

In this chapter, we intend to concentrate specifically on GFD over domains with similarities and the directly related issues. From the logical point of view, the generalization of FDs to FDs over domains with similarities may be looked at as replacing two-valued identity relations by many-valued ones which represent similarities. This step may be considered as switching from a two-valued logic, as the formal framework in which the ordinary model is implicitly developed [1], to appropriate fuzzy logic. Since two-valued

[1] Codd's original model was based on two-valued logic, although later Codd himself extended its rela-

logic may be considered as a particular case of a fuzzy logic, the logical viewpoint makes it naturally possible to reflect on whether and in what sense a particular approach to GFD over domain with similarities is a proper generalization of the ordinary one. The switch to fuzzy logic naturally brings the question of how the concept of validity, entailment etc. should be dealt with. Should the validity of a GFD in a given relation remains bivalent (true or false) or should the validity be many-valued (e.g. taking values from $[0, 1]$)? These are our "ideological roots": we are aware that other generalizations are possible than those that may be addressed from the logical viewpoint but we insist on the claim that good approaches need to have logical foundations to follow the idea of the relational model of data.

There are several works which addressed and examined the various proposals to GFDs. In [27] the authors focused on the semantics of various extensions of functional dependencies. In [28] the authors concentrate on the connection between GFD and redundancy elimination. In [21] it is shown that some approaches can be considered as a special case of the approach given by Belohlavek and Vychodil. Recently, in [111] a list of different kinds of approaches to functional dependencies in generalized relational model was presented. The first two works are almost twenty years old now and thus do not provide up to date information. More importantly none of the works is trying to unify various approaches or to objectively compare them.

We are going to focus on GFD over domains with similarities, in this chapter we want to: (i) present a reasonably complete list of various definitions of similarity-based GFDs and critically examine them, (ii) provide a unifying framework for different approaches using fuzzy logic based on complete residuated lattices, (iii) objectively compare them using our criterion (which is based on the notion of fuzzy function).

The rest of this chapter is organized as follows: Section 3.1 provides a first glimpse in the area of GFD and GRM. Section 3.2 contains an exhaustive presentation of similarity-based approaches to GFD which appear in the literature. Each approach is described by paying attention to the structure of truth degrees and to the extension of relation (data table). In Section 3.3 we introduce a widely accepted definition of fuzzy function as a basis for the development of our comparative criterion, which we further use for comparing the influential generalizations of functional dependencies. In Section 3.4 we summarize the results concerning different generalizations of FD and we give some conclusions.

3.1 Generalizations of the relational model

The main goal of this section is to look into generalizations of the relational model involving similarity relations. The extension of domains with a similarity relation usually does not

tional calculus by considering a three-valued logic to manage missing, non-applicable or unknown information via the NULL value [42]. In the further step [43] a four-valued logic was introduced to deal separately with these different types of uncertainties. Nevertheless, these extensions have been subject to criticism in the past (see C.J. Date in [50]).

stand alone but comes together with ranked data tables, and various data extensions. Therefore in the following overview we include works that are dealing with at least one of the following three issues: similarity on domains, ranked data tables, and data extension.

1. **Similarity-based approaches (from equality to similarity)**: In most of the approaches we will consider, domains are additionally equipped with some kind of similarity relation to denote the degrees of similarity between domain values. Thus the equality relation that is implicitly presented in the original Codd's model (domain values are either "equal" or "not equal") is replaced by a binary fuzzy relation that maps every pair of domain values to $[0, 1]$ and is meant to express the similarity (closeness) of domain values [26, 30, 31, 46, 35, 56, 92, 95, 96, 97, 116, 103] and later in [15, 25, 60, 76, 80, 99, 106, 122]. By similarity (also called resemblance or proximity) is usually meant a reflexive symmetric measure. Sometimes an additional property, namely transitivity, is required. Contrary, there are approaches in which similarity is defined as (only) reflexive relation [66]. Although the degree of similarity comes usually from $[0, 1]$, there are extensions considering more general algebraic structures, e.g. commutative semiring [66] or residuated lattice [15, 44].

2. **Rank-based approaches (from relation to fuzzy relation)**: By rank-based approaches we mean extensions of the relational model in which the data table is seen as a fuzzy set of tuples (in the original model the data table is simply a set of tuples). Thus the data table has an additional column which contains a rank—also called (membership) grade, score or weight—to express to what degree a tuple belongs to a data table. First attempts to rank-based approaches can be found in [4, 71, 95, 108, 123]. Later works include [21, 115, 59, 85, 87, 96, 104, 106]. There are also extensions in which the rank is assigned to every attribute value, e.g. in [83, 44, 22]. The ranks usually take values from $[0, 1]$, but there are approaches in which the unit interval is replaced by some general algebraic structure, e.g. commutative semiring [65], De Morgan frame [66], or residuated lattice [21]. In one of the pioneering work done by Umano [108] the rank itself is a possibility distribution on $[0, 1]$. The idea that the rank is a non-single value appears also in [87], where the rank is a pair of possibility and necessity measures to indicate the possibility and necessity that a tuple satisfies a certain constraint.

   The meaning of the rank differs among approaches and it is seen as:

   (a) Compatibility with the relation, e.g. in [4] the rank is "a degree to which t satisfies the relation or is compatible with the relation". Later in [28] the rank is understood as a "degree to which tuple belongs to the relation, which is then supposed to have a fuzzy (or gradual meaning)". For example consider relation `Young employee` with attributes `Name` and `Age`, then a tuple belongs to the data table to the degree to which it satisfies the concept "Young".

(b) Global confidence level, see [28] for example: "(The weight is a) global confidence level in information stored in the tuple which is a part of a relation representing all or nothing concept". Consider relation `Likes` with attributes `Name` and `Movie`, then the degree to which a tuple belongs to the relation is understood as the confidence in the information stored in the tuple, but with relation `Likes` remaining crisp.

(c) Compatibility with the set of individual constraints specified on the relation, see [85, 87].

(d) Degree to which a tuple matches a query, see [21, 59, 96], or the degree to which it is possible that a tuple matches a query, see [32].

The fact that the interpretation of the rank differs among approaches is puzzling, as it was already pointed out by Dubois and Prade in [56]. Moreover, there are approaches in which the meaning of the rank is not clearly explained, for example in [71].

3. **Data extensions (from precise to imprecise values)**:
The third aspect involved in the various generalizations of the relational model is data extensions, i.e. replacing precise values by imprecise ones. There are several approaches where the authors are trying to incorporate more complex data, namely an attribute value is considered to be a set of (possible) values in [31, 35, 101, 116, 113], a fuzzy set (including linquistic terms) or a possibility distribution in [26, 88, 48, 56, 35, 75, 77, 80, 95, 87, 93, 104, 108], a vague set in [122] or an interval-valued possibility distribution in [86]. When a (fuzzy) set is considered as an attribute value, it is important to know its interpretation. Any set can be interpreted in two different ways, either as a conjunctive set (also called ontic set) or as a disjunctive set (also called epistemic set). One set can be interpreted in both ways, for example consider the crisp set of Jane's pets: $\{dog, cat, parrrot\}$. In the ontic point of view the set represents the fact that Jane has three pets at home. By contrast, in the epistemic point of view, the set represents the fact that Jane has one pet at home—a dog or a cat or a parrot, where the or is exclusive. In this case the set is understood as a possibility distribution (all three values are equally possible) and represents uncertainty in our knowledge. This applies to fuzzy sets as well. On the one hand, fuzzy sets can represent gradual entities or linguistic variables (conjunctive meaning) and on the other hand fuzzy sets can be used as possibility distributions [120] (disjunctive meaning). This distinction was made by Zadeh in [121] and later discussed for example in [57]. When the attribute value is considered to be a set or a fuzzy set, it is crucial to know whether the meaning is conjunctive or disjunctive. Unfortunately, the interpretation is not always clear and does not usually affect the semantics of GFD. Nevertheless, in Codd's relational model there are no limitations in what can and cannot be an attribute value. C. J. Date, one of the leading experts on relational databases, pointed out that [52]: "... the domains over which relations

are defined can be of arbitrary complexity. As a consequence, we can have attributes of relations—or columns of tables, if you prefer—that contain geometric points, or polygons, or X rays, or XML documents, or fingerprints, or arrays, or lists, or relations, or any other kinds of values you can think of. But this idea too was always part of the relational model! The idea that the relational model could handle only rather simple kinds of data (like numbers and strings and dates and times) is a huge misconception, and always was. . . .".

We therefore argue that approaches considering only more complex data (without any further extensions of the model) should not be seen as a genuine extensions of the original relational model, even when they consider fuzzy sets as attribute values.

3.2 Survey of similarity-based generalizations of FD

In this section we are going to present an exhaustive set of approaches to similarity-based GFDs. To ease the reading of this section we point out some problems that need to be solved once the equality is replaced by a similarity relation.

The definition of classical functional dependency introduced in Equation (2.44) can be rewritten as follows:

$$||A \Rightarrow B||_{\mathcal{D}} = \min\{(r_1(A) = r_2(A)) \rightarrow (r_1(B) = r_2(B)) \mid r_1, r_2 \in \mathcal{D}\}. \tag{3.1}$$

In the above equation the equality is understood as bivalent (either two values from a domain are equal or not) and the implication is the classical one (taking values from the set $\{0, 1\}$). Again, the underlying logic of the Codd's relational model is the classic predicate logic. Although the following is obvious, we want to explicitly mention that in the original relational model: the truth value of FD, the degree to which an FD follows from a set of FDs, as well as the degree to which a tuple matches a query, come from the same set, $\{0, 1\}$. We are going to list generalizations of FD which replaced in the Equation (3.1) equality by similarity. As a consequence, the set $\{0, 1\}$ is replaced by some more general partially ordered set (usually $[0, 1]$) for expressing the similarity of two domain values. Having similarity relation on each domain many questions suddenly arise:

- *[AtrSim]* In the case of more complex data (e.g. attribute value is a set of possible values, or a fuzzy set): how should the similarity of attribute values be defined?
- *[TuplSim]* How to define the similarity of tuples based on the similarity of the corresponding attribute values?
- *[Imp]* What implication should be used? Note that now $r_1(A) \approx r_2(A)$, $r_1(B) \approx r_2(B)$ are degrees from the previously chosen set.
- *[TrGFD]* Should the notion of GFD remain crisp? This will mean that the GFD is either true or false in a given relation. Or should the concept become many-valued,

meaning that we allow the GFD to be satisfied to certain degree (between the two borderline cases: not satisfied, satisfied).

- *[Rank]* Having similarities on domains, how should the similarity-based queries be evaluated? Should some degree to which data match a similarity-based condition appear? In [59] Fagin said about querying in multimedia database systems that: "*...it is convenient to introduce "graded" (or "fuzzy") sets, in which scores are assigned to objects, depending on how well they satisfy atomic queries*". We believe this applies not only to multimedia databases but to all databases where similarities are involved.

We would like the reader to keep in mind these questions when going through the survey, the comparison based on how these questions are answered can be found in Section 3.4. We will now go through various attempts to generalize FD that use similarity measures and reformulate them using residuated lattice as a structure of truth degrees. We will not focus on how various similarity measures were defined (the *[AtrSim]* problem), although some important and widely cited approaches are mentioned, but mainly on how the similarity is used in the generalization of FDs.

If not otherwise stated we assume a relation scheme $R = \{y_1, \ldots, y_n\}$, $A, B \subseteq R$. Similarity or equivalence relation on domain D_i of attribute y_i will be denoted as $\approx_i$, $\equiv_i$, respectively. In most of the approaches that used the unit interval as a set of truth degrees, the *[TuplSim]* issue is solved by using the minimum of the similarities of the corresponding attribute values. If not otherwise stated we assume that the similarity of two tuples r_1, r_2 from some data table $\mathcal{D}$ on R is defined as:

$$r_1(A) \approx_{\mathcal{D}} r_2(A) = \min_{y_i \in A} r_1(y_i) \approx_i r_2(y_i). \tag{3.2}$$

If there is no confusion we will write $r_1(A) \approx r_2(A)$ instead of $r_1(A) \approx_{\mathcal{D}} r_2(A)$.

Buckles and Petry (1983): One of the pioneering work was done by Buckles and Petry, see [31]. The authors introduced a model, which will be later referred as Buckles-Petry model, where domains are equipped with fuzzy equivalence relations (called similarity in the original work) and tuple values are allowed to be (ordinary) non-empty subsets of the domain. That is,

$$\mathcal{D} \subseteq \prod_{y \in R} \underline{2}^{D_y}, \text{ where } \underline{2}^{D_y} \text{ denotes } 2^{D_y} \setminus \{\varnothing\}. \tag{3.3}$$

The authors called such relation $\mathcal{D}$ a fuzzy relation, although relation $\mathcal{D}$ is an ordinary relation, i.e. an ordinary subset (not fuzzy subset) of some cross product. Each domain is equipped with fuzzy equivalence, i.e. reflexive, symmetric and transitive relation, which maps every pair of domain values to $[0, 1]$. The transitivity was given by two different inequalities:

$$T1: \quad u \equiv_i w \geq \max_{v \in D_i} \{\min\{u \equiv_i v, v \equiv_i w\}\} \tag{3.4}$$

$$T2: \quad u \equiv_i w \geq \max_{v \in D_i} \{u \equiv_i v * v \equiv_i w\}, \tag{3.5}$$

where * is an arithmetic multiplication. The correspondence with $\otimes$-transitivity (2.42) is obvious when one considers Gödel and Goguen t-norm. As far as we know, it was the first time when some kind of similarity relation was used in database design.

In the earlier work [31] the authors considered domains which may consist of a finite (or infinite) set of scalars or a finite (or infinite) set of numbers with appropriate similarity relation. Later [32] the authors began to propound the idea that domains may also consist of linguistic values or fuzzy numbers (in a example given in [32] a fuzzy number was used to express preferences). The meaning of a set of values (used as an attribute value) is conjunctive, see [33] where the authors distinguished between their model (as an example of uniform data model) and possibilistic data models.

Remark 3. *In [32] the authors presented part of relational algebra for their generalized model and introduced the concept of a rank—query Q can induce a membership degree for every tuple $r \in \mathcal{D}$, denoted in the original work as $\mu_Q(r)$, which represents the "possibility of matching the query specifications". But this means that after executing a query and obtaining a ranked data table, we actually leave the model, because according to* (3.3) *a data table is an ordinary set. Put it in another way: the result of a query may not be a valid data table.*

Generalized functional dependencies (called fuzzy FD) were defined in [3]: Let $0 < \beta \leq 1$. The GFD $A \Rightarrow_\beta B$ holds in the Buckles-Petry model, iff for every pair of tuples $r_i = (d_{i1}, \ldots, d_{in}), r_j = (d_{j1}, \ldots d_{jn})$:

$$\min_{y_k \in A} \{ \min_{u \in d_{ik}, v \in d_{jk}} u \equiv_k v\} \leq \beta * \min_{y_r \in B} \{ \min_{u \in d_{ir}, v \in d_{jr}} u \equiv_r v\}, \tag{3.6}$$

where d_{ik} is the value of attribute y_k for tuple r_i and $*$ is the arithmetic product. In the above definition, β is a parameter which influences the validity of generalized functional dependency. Observe that if β is close to 0, the GFD will hardly be fulfilled in any table. Moreover, if a relation $\mathcal{D}$ satisfies classical functional dependency $A \Rightarrow B$, then it will satisfy the dependency given by Equation (3.6) if and only if $\beta = 1$ and all values are singletons, see Example 1.

Later, this definition was modified and reformulated using the so called conformance [116] and by moving the β parameter from the right hand side to the left hand side of (3.6). Thanks to this the classical FD can be captured by the GFD even for $\beta \neq 1$, but still under assumption that all attribute values are singletons. The conformance of attribute $y_k \in R$ for tuples $r_1, r_2 \in \mathcal{D}$, denoted as $C(y_k[r_1, r_2])$, is given by the following formula:

$$C(y_k[r_1, r_2]) = \min_{u,v \in d_{1k} \cup d_{2k}} u \equiv_k v. \tag{3.7}$$

Note that the conformance is not necessary reflexive, which leads to odd behavior as demonstrated in Example 1. Moreover, for $A \subseteq R$: $C(A[r_1, r_2]) = \min_{y_k \in A} C(y_k[r_1, r_2])$. The

	y_1	y_2	y_3
r_1	$\{a, b\}$	$\{c, d\}$	$\{c\}$
r_2	$\{a, b\}$	$\{c, d\}$	$\{d\}$

Table 3.1: Relation $\mathcal{D}$ in the Buckles-Petry model.

GFD $A \Rightarrow_\beta B$ is satisfied in the Buckles-Petry model if and only if for every pair of tuples r_1, r_2 we have:

$$\beta * C(A[r_1, r_2]) \leq C(B[r_1, r_2]), \tag{3.8}$$

where $\beta \in [0, 1]$ is called linguistic strength and is optional. The default value of β is 1.

Example 1. *Both* (3.6) *and* (3.8) *behave unnaturally. Assume* $R = \{y_1, y_2, y_3\}$ *with* $D_1 = D_2 = D_3 = \{a, b, c, d\}$ *and relation* $\mathcal{D}$ *from Table 3.1.*

First, note that the table can be seen as an ordinary relation in the Codd's relational model over domains $D_1' = \underline{2}^{D_1}$, $D_2' = \underline{2}^{D_2}$, $D_3' = \underline{2}^{D_3}$ *and it is in first normal form, since it is "a direct and faithful representation of some relation", see [53]. We can see that the classical FD* $\{y_1\} \Rightarrow \{y_2\}$ *is satisfied in* $\mathcal{D}$*. Now one would expect that the GFD* $\{y_1\} \Rightarrow \{y_2\}$ *holds in* $\mathcal{D}$ *for any* β *as well. But it is not the case. If* $a \equiv_1 b > c \equiv_2 d$*, then according to* (3.6) *the GFD does not hold for* $\beta = 1$*. If* $a \equiv_1 b$ *is much greater than* $c \equiv_2 d$*, then* β *must be close to* 0 *in order to make* $\{y_1\} \Rightarrow \{y_2\}$ *valid. The same remark holds when one takes* (3.8) *instead of* (3.6)*.*

The problem illustrated in Example 1 was solved in [117] by proposing a new definition of conformance:

$$\mathcal{C}(y_k[r_1, r_2]) = \min\{\min_{u \in d_{1k}}\{\max_{v \in d_{2k}}\{u \equiv_k v\}\}, \min_{u \in d_{2k}}\{\max_{v \in d_{1k}}\{u \equiv_k v\}\}\} \tag{3.9}$$

This definition of conformance yields to reflexive and symmetric measure and therefore we can employ our notation for similarity relations. Let us denote $C(A[r_1, r_2])$ by $r_1(A) \approx r_2(A)$. Since the $*$ from (3.8) is arithmetic product and the similarity takes values from $[0, 1]$, authors actually use the standard product algebra $[0, 1]_\Pi$ as structure of truth degrees (i.e. $\otimes$ and $\rightarrow$ are Goguen adjoint operations). Equation (3.8) can be formulated as follows:

$$\beta \otimes (r_1(A) \approx r_2(A)) \leq (r_1(B) \approx r_2(B)), \tag{3.10}$$

which is also equivalent to:

$$\beta \leq (r_1(A) \approx r_2(A)) \rightarrow (r_1(B) \approx r_2(B)). \tag{3.11}$$

As a consequence, the definition of GFD can be reformulated as follows:

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1, r_2 \in \mathcal{D}} \big((\beta \otimes r_1(A) \approx r_2(A)) \rightarrow r_1(B) \approx r_2(B)\big)^*, \tag{3.12}$$

where * is the globalization.

Shenoi at al. [100, 101] claimed to extend the Buckles and Petry model by considering a family of partitions on each domain (ordinary equivalence relation) instead of a fuzzy equivalence relation. Each partition is determined by a level of precision, denoted as α_i, which means that elements in the same equivalence class are "in relation to each other to a degree no lower than α_i".

The notion of data table remains the same as in Equation (3.3). To understand the defintion of GFD, we need one more concept: redundancy at level of partition. Two tuples r_1, r_2 are called redundant at level $\alpha = (\alpha_y)_{y \in R}$, denoted by $r_1 \sim_\alpha r_2$, iff for every $y \in R$, $r_1(y)$ and $r_2(y)$ are subsets of the same equivalence class in the partition $P_y(\alpha_y)$. A functional dependency $A \Rightarrow B$ holds with respect to partition levels $\alpha = (\alpha_y)_{y \in A}$ and $\beta = (\beta_y)_{y \in B}$ in $\mathcal{D}$ if, for every pair of tuples r_1, r_2, if they are redundant at level α for attributes in A, then they are redundant at level β for attributes in B. That is, $r_1(A) \sim_\alpha r_2(A)$ implies $r_1(B) \sim_\beta r_2(B)$. If we will use the notion of similarity for reformulation of GFD given by Shenoi et al., we will obtain (3.34). We are free to do that since in a subsequent paper [84], Shenoi et al. admit that there is no distinction between their model and Buckles and Petry's one: "*Our model was thought to generalize Buckles and Petry's work because tuple components in their model are always non-empty subsets of equivalence classes. However, ... our early work is essentially a reformulation of Buckles and Petry's work.*" In [84] the authors proposed so called *complete-lattice-equivalence-class model*, in which each domain is associated with a complete lattice of (ordinary) equivalence relations (and thus partitions). The size and structure of each complete lattice of equivalence relations is required to be the same, more precisely to be isomorphic to previously given lattice L. The lattice L becomes part of a relation scheme.

Prade and Testemale(1984): In [93] Prade and Testemale considered so called possibilistic fuzzy data model, i.e. attribute values are allowed to be possibility distribution in Zadeh's sense [120]. A model based on the concept of possibility distribution was originally proposed by Umano [108]. The model of Prade and Testemale is a slight generalization of the Umano concept by introducing an extra element denoted by e, which is used when there is a nonzero possibility that the attribute does not apply. The relation $\mathcal{D}$ is defined as:

$$\mathcal{D} \subseteq \prod_{y \in R} [0,1]^{D_y \cup \{e\}}, \tag{3.13}$$

where $[0,1]^{D_y \cup \{e\}}$ denotes the set of all possibility distributions on $D_y \cup \{e\}$. Moreover, each domain $D_y \cup \{e\}$ is associated with a similarity relation (called fuzzy proximity relation) $\sim_y$ which takes values from $[0,1]$. The similarity relation is then extended to possibility distributions on $D_y \cup \{e\}$ as follows: For $r_1(y), r_2(y) \in [0,1]^{D_y \cup \{e\}}$:

$$r_1(y) \approx_y r_2(y) = \max_{u,v \in D_y \cup \{e\}} \min\{u \sim_y v, (r_1(y))(u), (r_2(y))(v)\}. \tag{3.14}$$

$r_1(y) \approx_y r_2(y)$ is the possibility that values $r_1(y)$ and $r_2(y)$ are similar in the sense of $\sim_y$.

If all possibility distributions are normal [2], then $\approx_y$ is a similarity relation for all $y \in R$. The GFDs were introduced only for singleton sets. Given a fixed threshold $\lambda \in [0,1]$ and $y_i, y_j \in R$, the GFD $\{y_i\} \Rightarrow \{y_j\}$ is satisfied in $\mathcal{D}$ if and only if for all $r_1, r_2 \in \mathcal{D}$

$$(r_1(y_i) = r_2(y_i)) \rightarrow (r_1(y_j) \approx_j r_2(y_j) \geq \lambda), \tag{3.15}$$

where $\rightarrow$ is the ordinary implication. The FD should capture the following: "If the values of the attribute y_i are equal for r_1 and r_2, we may want to express that the values of the attribute y_j for r_1 and r_2 cannot be far from each other".

This definition can be extended to sets of attributes and reformulated as follows: Assume $L = [0,1]$ and * being globalization. Then for any t-norm and corresponding residuum:

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1, r_2 \in \mathcal{D}} \big((r_1(A) = r_2(A)) \rightarrow (\lambda \rightarrow r_1(B) \approx r_2(B))^*\big), \tag{3.16}$$

where $r_1(B) \approx r_2(B)$ is given by (3.2). Moreover, if the $\approx_y$'s are separable, then we can use $(r_1(A) \approx r_2(A))^*$ instead of $r_1(A) = r_2(A)$.

Remark 4. *The authors introduced part of relational algebra in [93]. The result of a query consists in general of two fuzzy sets: the set of tuples which possibly satisfy the condition and the set of tuples which necessarily satisfy the condition. This again means that the result of a query is not in correspondence with* (3.13)*.*

Remark 5. *Nakata in [87] used the similarity given by Equation* (3.14) *to define the compatibility of a relation with a functional dependency. The compatibility itself is a pair of possibility and necessity measures. Given $y_i, y_j \in R$ with $y_i \neq y_j$ and assuming $a \rightarrow b = \neg a \vee b$, the possibility that a given relation $\mathcal{D}$ is compatible with an FD $\{y_i\} \Rightarrow \{y_j\}$ is defined as*

$$\bigwedge_{r_1, r_2 \in \mathcal{D}, r_1 \neq r_2} \max\{\neg(r_1(y_i) \approx_i r_2(y_i)), r_1(y_j) \approx_j r_2(y_j)\}, \tag{3.17}$$

where $\neg(r_1(y_i) \approx_i r_2(y_i)) = \max\limits_{u,v \in D_i} \min\{1 - (u \sim_i v), (r_1(y_i))(u), (r_2(y_i))(v)\}$. The necessity measure is then computed from possibility measure replacing $r_1(y_i) \approx_i r_2(y_i)$ by $1 - (\neg(r_1(y_i) \approx_i r_2(y_i)))$.

Raju and Majumdar (1988): Another generalization of FD was proposed by Raju and Majumdar [95]. They considered similarity relation on each domain and ranks associated to each tuple, but the ranks and similarity degrees come from [0.1]. More precisely, a relation $\mathcal{D}$ is a fuzzy subset on Tupl(R):

$$\mathcal{D} : \prod_{y \in R} D_y \rightarrow [0,1]. \tag{3.18}$$

[2] A possibility distribution $\pi \in [0,1]^X$ is *normal* if there is an element $x \in X$ such that $\pi(x) = 1$.

Therefore every tuple r has associated a degree (rank) to which the tuple belongs to $\mathcal{D}$, denoted as $\mathcal{D}(r)$. The meaning of the ranks in not clearly given. In Example 3.1 in [95] the authors say that a rank can be interpreted as a possibility measure, fuzzy measure of the association between values, or as a truth degree of a fuzzy predicate associated with given relation.

Depending on the complexity of domains, the authors classified their model into two categories, namely

- Type-1, where each domain may be a classical set or a fuzzy set. In this case, attribute values are singletons, taken from some set U.
- Type-2, where each domain may be a set of fuzzy sets or a set of possibility distributions (page 136 in [95]).

The authors considered both interpretations of a fuzzy set and they also provided two different interpretation of a rank (page 138 in [95]): "...as a possibility measure of association among the data or as a truth value of a fuzzy predicate associated with (relation) r."

Later, we will refer to the model where attribute values are allowed to be fuzzy sets and data table is understood as in (3.18) as Raju-Majumdar's model. The generalized functional dependency $A \Rightarrow B$ is satisfied by a relation $\mathcal{D}$ iff for all $r_1, r_2 \in \mathcal{D}$ ($r_1, r_2 \in \mathcal{D}$ means $r_1, r_2 \in \mathrm{Tupl}(R)$ with $\mathcal{D}(r_1) > 0$ and $\mathcal{D}(r_2) > 0$)

$$r_1(A) \approx r_2(A) \leq r_1(B) \approx r_2(B). \tag{3.19}$$

The inequality can be reformulated using Rescher-Gaines (RG) implication, $a \to_{RG} b = 1$ iff $a \leq b$, 0 otherwise,

$$r_1(A) \approx r_2(A) \to_{RG} r_1(B) \approx r_2(B). \tag{3.20}$$

This reformulation often appears in the literature, but RG implication is not a residuated implication and therefore we prefer the following formulation: For $L = [0, 1]$, any t-norm and corresponding residuum, and for hedge being globalization the definition of GFD given by Equation (3.19) is equivalent to:

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1, r_2 \in \mathcal{D}} \big(r_1(A) \approx r_2(A) \to r_1(B) \approx r_2(B)\big)^*. \tag{3.21}$$

Remark 6. *Some inconveniences arise from this definition: (i) First, note that the rank is not involved in the definition of GFD. As a result tuples with low degrees of membership can significantly influence the validity of GFD.*
(ii) Furthermore, note that if $r_1(A) \approx r_2(A) = 1$, then $r_1(B) \approx r_2(B)$ must also be equal to 1, in order to satisfy GFD given by (3.19)*. This behavior is seen as a weak spot of the definition and was mentioned by several authors. In [97] Saharia and Barron addressed this problem and introduced cluster dependencies to solve this issue.*
(iii) The completeness of inference axioms is conditioned by the following: For each domain

D_i there is at least one pair $u, v \in D_i$ such that $u \approx_i v = 0$. It was later shown by Belohlavek and Vychodil in [21] that this condition is not needed. Furthermore, Belohlavek and Vychodil proved that although the semantics of Raju-Majumdar model is richer, we can not infer anything new comparing to the original Codd's model. More precisely: An FD $A \Rightarrow B$ follows from a set T of another FDs in the sense of Raju-Majumdar iff $A \Rightarrow B$ follows from T in the sense of ordinary Codd's model.

The paper by Raju and Majumdar is probably the most influential one and the definition of the GFD given by (3.19) inspired many authors. The following authors used the same definition of GFD as Raju and Majumdar, but presented a new notion of similarity, or a new extension of Codd's model:
(i) In [80] the authors followed the model of Raju-Majumdar. But the similarity of tuple values was defined using a notion of semantic space.
(ii) In [122] the authors presented a GFD in the framework of so called vague relational database, i.e. databases where tuple values are allowed to be vague sets.
(iii) Also Wei-Yi Liu in [79] used the same idea for definition of GFD. However, the author considered intervals as fuzzy attribute values and used "Semantic proximity (SP)" instead of similarity in the definition of GFD. SP is a symmetric measure, but it is not reflexive in general. The author also presented Armstrong's axioms and claimed them to be sound and complete. Unfortunately, the non-reflexivity of semantic proximity produces a mistake as pointed out in [49], where the authors showed that two of the inference rules given by Liu are not sound. The soundness is guaranteed by reflexivity of SP.
(iv) Later, in [75] a new semantic proximity was defined for intervals, which is again not reflexive. (v) Liu also used semantic distance instead of semantic proximity to define the GFD, see [78, 77].
(vi) The work done by Raju and Majumdar inspired also Saxena and Tyagi. In [98] the authors used a special fuzzy set ϕ to represent null value "does not apply". The null value unknown is represented differently. This approach differs from its precursors due to the specific treatment of the special fuzzy set ϕ. a GFD $A \Rightarrow B$ holds in $\mathcal{D}$ according to Saxena and Tyagi if for all pair of tuples r_1, r_2 such that $\mathcal{D}(r_1) > 0$, $\mathcal{D}(r_2) > 0$, $r_1(y) \neq \phi \neq r_2(y)$, for each $y \in A$, and $r_1(A) \approx r_2(A) > 0$, one of the following conditions holds:

1. $r_1(B) = r_2(B) = \phi$, or
2. there exists a nonempty set $B' \subseteq B$ such that $r_1(y) \neq \phi \neq r_2(y)$ for each $y \in B'$, $r_1(B \setminus B') = r_2(B \setminus B') = \phi$ and $r_1(A) \approx r_2(A) \leq r_1(B') \approx r_2(B')$.

Chen (1991): Another significant proposal of definition of GFD was developed by Chen [39], see also [36, 35]. Chen used the possibilistic fuzzy data model:

$$\mathcal{D} \subseteq \prod_{y \in R} [0,1]^{D_y}, \tag{3.22}$$

where $[0,1]^{D_y}$ denotes the set of all possibility distributions over domain D_y. Moreover, a similarity relation $\sim_y$ (originally called closeness relation) is associated with each do-

main D_y, which is then used to express the similarity $\approx_y$ of attribute values (possibility distribution). The GFD $A \Rightarrow B$ holds in $\mathcal{D}$ to degree θ iff

$$\min_{r_1,r_2 \in \mathcal{D}} (r_1(A) \approx r_2(A) \rightarrow_I r_1(B) \approx r_2(B)) \geq \theta, \tag{3.23}$$

where $\rightarrow_I$ stands for "fuzzy implication operator", i.e. $\rightarrow_I$: $[0,1] \times [0,1] \rightarrow [0,1]$ and satisfies for every $a, b, c \in [0,1]$:

$$\begin{aligned} &a \rightarrow_I b = 1 \text{ iff } a \leq b, \\ &a \rightarrow_I b \leq \min(a,c) \rightarrow_I \min(b,c), \\ &\min(a \rightarrow_I b, b \rightarrow_I c) \leq a \rightarrow_I c. \end{aligned}$$

The GFD expresses the fact that: "Close B values correspond to close A values". Later, Chen et al. [61] have proposed a specific form of the previous definition in order to express the fact that "Close B values correspond to close A values, and identical B values correspond to identical A values". The $\rightarrow_I$ is classical implication when $r_1(A)$ and $r_2(A)$ are identical, and Gödel implication otherwise, i.e. the GFD $A \Rightarrow B$ holds in $\mathcal{D}$ to a degree θ iff for all pair of tuples r_1, r_2:

$$\begin{aligned} &\text{if } r_1(A) = r_2(A) \text{ then } r_1(B) = r_2(B), \\ &(r_1(A) \approx r_2(A) \rightarrow_G r_1(B) \approx r_2(B)) \geq \theta \text{ otherwise.} \end{aligned} \tag{3.24}$$

By using Gödel implication in the second part of the definition, the meaning of the fact that GFD $A \Rightarrow B$ is true to degree θ is as follows: For each pair of tuples: the similarity on attributes B is at least as high as the similarity on attributes A or greater then θ. The last inequality can be reformulated as follows:

$$\theta \otimes r_1(A) \approx r_2(A) \leq r_1(B) \approx r_2(B). \tag{3.25}$$

Note the correspondence with Equation (3.10), but now $\otimes, \rightarrow$ are Gödel operations. For $L = [0,1]_G$ being the standard Gödel algebra, the GFD given by (3.24) can be reformulated as follows:
$||A \Rightarrow B||_{\mathcal{D}} = \theta$ if

$$\theta \leq \bigwedge_{r_1,r_2 \in \mathcal{D}} \big(r_1(A) = r_2(A) \rightarrow r_1(B) = r_2(B)\big) \wedge (r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B)). \tag{3.26}$$

The fact that $||A \Rightarrow B||_{\mathcal{D}} = \theta$ does not exclude existence of other $\theta' > \theta$ for which the inequality (3.26) holds. The soundness and completeness of Armstrong-like inference rules has been proved in [36]. Chen et al. also proposed normal forms, whose definitions remain the same as the classical ones with the notion of FD replaced by the notion of author's GFD, see [38, 37].

Remark 7. *In [112] the authors presented an algorithm for mining GFD based on Chen's definition. The authors first transform quantitative data into fuzzy data (e.g. the value*

10 000 $ for `salary` *attribute is modified into the value 0.3/Low Salary) and then search for GFD. In [102] authors considered* (3.23) *and demonstrated on particular examples how different similarity measures and different implications (Łukasiewicz, Gödel, etc) influence the result. The definition of GFD given by Equation* (3.24) *was recently used in the framework of interval-valued possibility distribution, see [86].*

Bhuniya and Niyogi (1993): According to Bhuniya and Niyogi [26] the generalized functional dependency $A \Rightarrow B$ holds in a Raju-Majumdar's model (3.18) if and only if for all $r_1, r_2 \in \mathcal{D}$ one of the following conditions holds

$$r_1(A) \approx r_2(A) \leq r_1(B) \approx r_2(B), \tag{3.27}$$

$$r_1(A) \approx r_2(A) - r_1(B) \approx r_2(B) \leq 1 - \beta, \tag{3.28}$$

where $r_1(A) \approx r_2(A) \geq \alpha$, $r_1(B) \approx r_2(B) \geq \alpha$, and $\alpha < \beta < 1$. In another words: if $\alpha \leq r_1(A) \approx r_2(A)$ and $\alpha \leq r_1(B) \approx r_2(B)$ then either (3.27) or (3.28). Or equivalently: $A \Rightarrow B$ holds in $\mathcal{D}$ iff for all $r_1, r_2 \in \mathcal{D}$ at least one of the following conditions holds:

$$r_1(A) \approx r_2(A) < \alpha \quad \text{or} \quad r_1(B) \approx r_2(B) < \alpha \quad \text{or} \quad (3.27) \quad \text{or} \quad (3.28). \tag{3.29}$$

First, note that $r_1(A) \approx r_2(A) < \alpha$ implies either $r_1(B) \approx r_2(B) < \alpha$ or (3.27). Therefore condition (3.29) is equivalent to

$$r_1(B) \approx r_2(B) < \alpha \quad \text{or} \quad (3.27) \quad \text{or} \quad (3.28). \tag{3.30}$$

Now, since $\beta < 1$, condition (3.27) implies condition (3.28) and therefore the disjunction "(3.27) or (3.28)" is equivalent to

$$\beta \leq r_1(A) \approx r_2(A) \rightarrow_{\mathrm{L}} r_1(B) \approx r_2(B) \tag{3.31}$$

in the standard Łukasiewicz algebra. As a consequence, $A \Rightarrow B$ holds (to degree 1) in $\mathcal{D}$ iff $\alpha \leq r_1(B) \approx r_2(B)$ implies $\beta \leq (r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B))$, for all $r_1, r_2 \in \mathcal{D}$. For hedge being globalization and for the standard Łukasiewicz algebra as a structure of truth degrees the following definition of GFD is equivalent to definition given by Bhuniya and Niyogi:

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1, r_2 \in \mathcal{D}} (\alpha \rightarrow r_1(B) \approx r_2(B))^* \rightarrow (\beta \rightarrow (r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B)))^*. \tag{3.32}$$

Cubero et al. (1994): Cubero et al. [48] proposed the following definition of an GFD for a possibilistic fuzzy data model (3.22). Each domain D_y is equipped with similarity relation (called proximity in the original work) $\sim_y$ and a fixed threshold c_y. GFD $A \Rightarrow B$ is satisfied iff for all $r_1, r_2 \in \mathcal{D}$:

$$(r_1(A) \approx r_2(A) \geq \alpha) \rightarrow (r_1(B) \approx r_2(B) \geq \beta). \tag{3.33}$$

Put into words: If $r_1(A)$ and $r_2(A)$ are similar at least to degree α, then $r_1(B)$ and $r_2(B)$ must be similar at least to degree β. Since $\rightarrow$ is classical implication, as long as $(r_1(B) \approx r_2(B)) \geq \beta$, it does not matter to what values $r_1(B)$ and $r_2(B)$ are associated with. The parameters α and β are vectors, $\alpha = (c_y)_{y \in A}$, $\beta = (c_y)_{y \in B}$, where values $c_y \in [0,1], y \in R$, are fixed and common to all GFDs. Thus $r_1(A) \approx r_2(A) \geq \alpha$ means $r_1(y) \approx_y r_2(y) \geq c_y$ for all $y \in A$. The definition of GFD given by Equation (3.33) can be reformulated as follows: Let $\mathbf{L}$ be any complete residuated lattice with universe $L = [0,1]$ and with globalization as a hedge:

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1, r_2 \in \mathcal{D}} \Big((\bigwedge_{y \in A} c_y \rightarrow r_1(y) \approx r_2(y))^* \rightarrow (\bigwedge_{y \in B} c_y \rightarrow r_1(y) \approx r_2(y))^* \Big). \quad (3.34)$$

Remark 8. *Later in [46] the authors used two different similarity measures for computing the similarity of tuple values (possibility distributions) in the antecedent and consequent part of GFD. In the antecedent part they used* (3.14)*, but for the consequent part they used the following:*

$$(r_1(y) \approx_y^{'} r_2(y)) = \min_{u,v \in D_y} \max\{u \sim_y v, 1 - (r_1(y))(u), 1 - (r_2(y))(v)\}. \quad (3.35)$$

The definition of GFD remained almost the same, more precisely: A GFD $A \Rightarrow B$ is satisfied in relation $\mathcal{D}$ iff (i) every tuple value is normalized, (ii) $r_i(y) \approx_y^{'} r_i(y) \geq c_y$ for every $y \in B$ and $r \in \mathcal{D}$, and (iii)

$$(r_1(A) \approx r_2(A) \geq \alpha) \rightarrow (r_1(B) \approx^{'} r_2(B) \geq \beta).$$

This definition of GFD was then used to define so called rule-based fuzzy functional dependencies [46, 47].

Remark 9. *A very similar idea appeared later in [2], where an FD captures the following: For every pair of tuples: If $r_1(A)$ and $r_2(A)$ are close to each other, then $r_1(B)$ and $r_2(B)$ must also be close to each other, more precisely:*

$$\forall r_1, r_2 \in \mathcal{D} \text{ If } \forall y_i \in A : |r_1(y_i) - r_2(y_i)| \leq \epsilon, \text{ then } \forall y_j \in B : |r_1(y_j) - r_2(y_j)| \leq \epsilon. \quad (3.36)$$

Ben Yahia et al. (1999): Ben Yahia, Ounalli, and Jaoua presented their definition of so called dynamic functional dependency in [115]. The word dynamic is used to emphasize the fact that an GFD can be true to some degree. The authors considered the Raju-Majumdar's model with uncertain data (fuzzy sets) and ranks coming from $[0,1]$. The dynamic FD is defined as follows: A determines B to degree β, denoted as $A \sim>_\beta B$, $\beta, \theta \in [0,1]$ in $\mathcal{D}$ if for all tuples r_1 and r_2 we have:

$$(r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B)) \geq \theta, \quad (3.37)$$

where

$$\beta = \min_{r_1, r_2}(r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B)), \quad (3.38)$$

and $\rightarrow$ is the Łukasiewicz implication. The threshold θ is fixed by the database designer. Note that the definition of Raju and Majumdar (3.19) is a special case for $\theta = 1$. The authors also proposed inference axioms and proved their soundness. The completeness is not proved.
The definition can be reformulated as follows: For $\mathbf{L} = [0,1]_{\text{Ł}}$ being the standard Łukasiewicz algebra, if $\bigwedge_{r_1,r_2\in\mathcal{D}} \big(r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B)\big) \geq \theta$, then

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1,r_2\in\mathcal{D}} \big(r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B)\big) \tag{3.39}$$

and $||A \Rightarrow B||_{\mathcal{D}} = 0$ otherwise.

Bosc et al. (1999): Another generalization was done by Bosc, Pivert, and Ughetto, see [30]. They were the first that used residuated implication corresponding to some t-norm. The authors worked with crisp data and similarity relation on every domain. Two generalizations of classical FD were proposed. Firstly, they relaxed the equality in the consequent only and secondly, they replaced the equality by similarity in both parts of the implication:

- Similarity is used only in the consequence part (which is meant to express tolerance) and GFD is defined as:

  $$\forall r_1, r_2 \in \mathcal{D} : r_1(A) = r_2(A) \rightarrow r_1(B) \approx r_2(B). \tag{3.40}$$

  Note the correspondence with the definition given by Equation (3.15). However, there is a big conceptual difference: The GFD given by Equation (3.15) remains bivalent (either it is true or not), whereas the GFD given above can be true to any degree from $[0,1]$.

- Similarity relation is used in both parts,

  $$\forall r_1, r_2 \in \mathcal{D} : r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B). \tag{3.41}$$

  Meaning: "The closer the A values, the closer the B values". For example: "Employees with similar experiences and jobs must have similar salaries." [30].

The reformulation using complete residuated lattice with universe $[0,1]$ is straightforward:

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1,r_2\in\mathcal{D}} (r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B)). \tag{3.42}$$

Unfortunately, the authors presented only those definitions and did not go any further by showing properties of such FD or presenting inference rules.

Tyagi et al. (2005): Later Tyagi et al. [106] introduced another generalization of functional dependencies using the framework of so called fuzzy functions. The authors developed GFD for Raju and Majumdar's model. Contrary to Raju and Majumdar,

$[0,1]_G$-equality (see Section 2) is employed in the model instead of similarity relation. The authors considered relation $\equiv_y$ on each domain which is separable (therefore reflexive), min-transitive and weakly symmetric (i.e. $(u \equiv v) = 1$ iff $(v \equiv u) = 1$ for all $u, v \in D_y$).

This approach was inspired by the definition of fuzzy function (see Definition 6 in Section 3.3) provided by Gottwald [63], which was later used and extensively studied by Demirci, see [55] for example. Relation $\mathcal{D}$ satisfies the GFD $A \Rightarrow B$ if its projection over $A \cup B$ (denoted as $\mathcal{D}_{AB}$) is a partial fuzzy function. That is, if $\forall r_1, r_2 \in \mathrm{Tupl}(A \cup B)$:

$$(\mathcal{D}_{AB}(r_1) \wedge \mathcal{D}_{AB}(r_2) \wedge r_1(A) \equiv r_2(A)) \leq r_1(B) \equiv r_2(B), \tag{3.43}$$

where $\mathcal{D}_{AB}(r) = \sup\{\mathcal{D}(r') \mid r' \in \mathrm{Tupl(R)} \text{ such that } r'(A \cup B) = r\}$.

The Definition (3.43) is a generalization of (3.19) in the sense that if the GFD is true according to (3.19) then it is also true according to (3.43). This approach has an advantage, which lies in the fact that the rank is involved in the definition of GFD. Assume that there is a pair of tuples violating $r_1(A) \equiv r_2(A) \leq r_1(B) \equiv r_2(B)$. In the case of (3.19) the GFD will be violated regardless of the ranks of these two tuples, but in the case of (3.43) the GFD may still be satisfied if the ranks are low enough. In general, we can say that the lower the rank the lower the influence on the validity of GFD. This is, in our opinion, the way how the GFD should behave when ranks are presented. If tuples have zero ranks (tuples do not belong to the relation), they should not influence the validity of GFD at all. Even this definition of GFD can be reformulated using complete residuated lattices. For $\mathbf{L}$ being the standard Gödel algebra (i.e. $a \otimes b = a \wedge b = \min(a, b)$) equipped with globalization we have:

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1, r_2 \in \mathrm{Tupl(A \cup B)}} \big((\mathcal{D}_{AB}(r_1) \otimes \mathcal{D}_{AB}(r_2) \otimes (r_1(A) \equiv r_2(A))) \rightarrow (r_1(B) \equiv r_2(B))\big)^*. \tag{3.44}$$

Kiss (1991): The idea that the rank should influence the validity of FD can be already found in [71]. Therefore we decided to include this approach here although the similarity relation is not employed in this model. The author considered ranks from $[0, 1]$ and presented the following Horn-formula of the first order logic:

$$\forall r_1, r_2 : (D(r_1) \wedge D(r_2) \wedge r_1(A) = r_2(A)) \Rightarrow r_1(B) = r_2(B). \tag{3.45}$$

When giving the semantic meaning to logical connectives, Kiss substituted $\wedge, \forall$ with the operator inf; $\vee, \exists$ with sup; $\Rightarrow$ with Łukasiewicz implication; and $\neg a = 1 - a$ for all $a \in [0, 1]$. The truth value to which the fuzzy relation $\mathcal{D}$ satisfies a given FD was given by

$$||A \Rightarrow B||_{\mathcal{D}} = 1 - \sup\{\inf(\mathcal{D}(r_1), \mathcal{D}(r_2)) |\, r_1(A) = r_2(A) \text{ and } r_1(B) \neq r_2(B))\}. \tag{3.46}$$

It is clear from (3.46) that the higher the degree of $\mathcal{D}(r_1)$ and $\mathcal{D}(r_2)$ when r_1, r_2 violate the classical FD, the lower the the truth degree of FD $A \Rightarrow B$.

The reformulation of (3.45) using residuated lattice is straightforward. For **L** being $[0,1]_{\mathrm{L}}$:

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1,r_2 \in \mathcal{D}} (D(r_1) \wedge D(r_2) \wedge r_1(A) = r_2(A)) \rightarrow r_1(B) = r_2(B). \tag{3.47}$$

Remark 10. *When going from* (3.45) *to* (3.46) *Kiss used the following rule:* $\neg(a \rightarrow b) = a \wedge \neg b$*, which does not hold in general when one takes Łukasiewicz implication and conjunction given by infimum. But the equality holds when* b *is 0 or 1, which is also our case, because* b *represents* $r_1(B) = r_2(B)$*.*

Belohavek and Vychodil (2006): Another extension is the proposal made by Belohavek and Vychodil, see [15]. We have already presented the definition of GFD, called similarity-based functional dependencies (SBFD) in Section 2.3.3. We will repeat the definition here for completeness of this survey. For $A, B \in L^R$ the degree to which SBFD $A \Rightarrow B$ is true in relation $\mathcal{D}$ over R is defined as:

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1,r_2 \in \mathrm{Tupl}(R)} \Big((r_1(A) \approx r_2(A))^* \rightarrow (r_1(B) \approx r_2(B)) \Big),$$

where

$$r_1(A) \approx_{\mathcal{D}} r_2(A) = (\mathcal{D}(r_1) \otimes \mathcal{D}(r_2)) \rightarrow \bigwedge_{y \in R} (A(y) \rightarrow r_1(y) \approx_y r_2(y)).$$

When compared to other approaches, the SBFD is true to some degree, which comes from the complete residuated lattice with hedge, and ranks are influencing the degree to which SBFD holds. Also note that contrary to previous approaches A, B are fuzzy sets.

Cordero et al. (2011): The last extension we want to mention in this section was presented by Cordero et al. in [45]. The authors worked with a generalization of Codd's relational model called fuzzy attribute table. The basic idea is that tuple value has assigned a rank coming from complete residuated lattice. More precisely, the fuzzy attribute table is understood as a map

$$\mathcal{D}\colon \prod_{y \in R} D_y \rightarrow L^R. \tag{3.48}$$

This means that for each tuple r: $\mathcal{D}(r) \in L^R$, i.e. $\mathcal{D}(r)$ is a tuple of truth values. For all $y \in R$, $\mathcal{D}(r)(y)$ is the truthfulness of tuple r in the value $r(y)$.

Remark 11. *Later in [22] Belohlavek and Vychodil introduced a similar model called Multi Ranked Data Tables where ranks (degrees) come from similarity-based queries and provided a relational algebra for this model.*

The definition of the GFD is accompanied with a Pavelka-style logic [89, 90, 91] called "Simplification Logic for fuzzy functional dependencies". The completeness is proved for a particular case of truth degrees, the unit interval. The authors introduced the following

definition: a fuzzy attribute table $\mathcal{D}$ is said to satisfy a generalized functional dependency $A \Rightarrow B$ with θ degree iff

$$\theta \leq \bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} (r_1(A) \approx_{\mathcal{D}} r_2(A)) \rightarrow (r_1(B) \approx_{\mathcal{D}} r_2(B)), \tag{3.49}$$

where $\rightarrow$ is a residuated implication. The similarity relation is called relative similarity in order to emphasize that the similarity of two tuples r_1, r_2 on the set of attributes $A \subseteq R$ depends on ranks:

$$(r_1(A) \approx_{\mathcal{D}} r_2(A)) = \bigwedge_{a \in A} \big((\mathcal{D}(r_1)(a) \otimes \mathcal{D}(r_2)(a)) \rightarrow (r_1(a) \approx r_2(a))\big). \tag{3.50}$$

In [45] the authors considered supremum of degrees to which the GFD is true. That is:

$$||A \Rightarrow B||_{\mathcal{D}} = \sup\{\theta \in [0,1] \mid \theta \text{ satisfies } (3.49)\}.$$

The last two approaches are connected with each other in the sense that the validity of GFD given by (3.49) may be expressed using the validity of GFD given by (2.62) and vice versa. A fuzzy attribute table $\mathcal{D}\colon \prod_{y \in R} D_y \rightarrow L^Y$ may be represented using a ranked data table $\mathcal{D}_R$ over the following family of domains $(\mathcal{D}_R)_y = D_y \times L$ (D_y are the original domains for $\mathcal{D}$) and mapping all the tuples to 1. More precisely: the ranks are defined by $\mathcal{D}_R(r) = 1$ if there exists $y \in R$ such that $\mathcal{D}(r)(y) \neq 0$, and $\mathcal{D}_R(r) = 0$ otherwise. Similarities are defined by

$$\langle d_1, a_1 \rangle \approx_y \langle d_1, a_2 \rangle = (a_1 \otimes a_2) \rightarrow \rho_y(d_1, d_2),$$

where ρ_y is the original similarity for $\mathcal{D}$. Conversely, a ranked data table $\mathcal{D}\colon \prod_{y \in R} D_y \rightarrow L$ can be transformed into a fuzzy attribute table $\mathcal{D}_F : \prod_{y \in R} D_y \rightarrow L^Y$ by considering for each tuple $r \in \mathcal{D}$: $\mathcal{D}_F(r)(y) = \mathcal{D}(r)$ for all $y \in R$.

Lemma 5. *For $A, B \subseteq R$, a fuzzy attribute data table $\mathcal{D}$ satisfies $A \Rightarrow B$ in degree $\theta \in [0,1]$ according to* (3.49) *iff $||A \Rightarrow B||_{\mathcal{D}_R} \geq \theta$ according to* (2.62)*. Therefore, $||A \Rightarrow B||_{\mathcal{D}} = ||A \Rightarrow B||_{\mathcal{D}_R}$.*

We have seen how to reformulate various definitions of GFD using complete residuated lattice as a structure of truth degree. It can be easily seen that the approaches given by (2.62), (3.49) as well as by (3.41) [3] are the most general ones, leaving other approaches as their particular cases. Discussion on this topic as well as proofs can be found in [21], where the authors showed that approaches given by (3.15), (3.19), (3.23) and (3.37) are special cases of (2.62). To summarize the results of the paper [21] note that: 1) Many approaches consider one particular case of t-norm (and corresponding implication), whereas GFD given by Equations (2.62) (this result applies to (3.49) and (3.41)as well) is developed for any t-norm. 2) Since A, B from (2.62) are in general fuzzy sets, the definition of GFD

[3] As we have already mentioned the promising and very general definition given by (3.41) was not developed any further.

given by (2.62) can easily incorporate GFD which use some additional parameters. For example for GFD given by (3.34) consider fuzzy sets A, B defined as: $A(y) = c_y$ for $y \in A$, $B(y) = c_y$ for $y \in B$.

Remark 12. ***Functional dependencies in terms of fuzzy rules***
In most of the previous approaches, FD are generalized by replacing the equality by similarity and by using concrete (not residuated in general) implication. Further, there are approaches in which different techniques are employed, but the resulting dependencies are still called fuzzy functional dependencies. Here we comment on some of them:
(i) Rasmussen and Yager [96] utilized linguistic summaries to define FDs. The authors considered a similarity relation defined on each domain and crisp data. FD can be in general true to some degree and should express the following: "If any two objects in the database have similar values for A then they have similar values for B." But the technique is completely different—the degree of satisfaction is computed for each tuple and then an average is taken.
(ii) Dubois and Prade in [56] and later Dubois, Prade and Bosc in [28] suggested to use fuzzy rules for definition of new kind of generalized functional dependencies (also called fuzzy functional dependencies) in the possibilistic fuzzy data model. Authors used certainty rules (the more x is A, the more certain y lies in B), possibility rules (the more x is A, the more possible B is a range for y), and gradual rules (the more x is A, the more y is B) and employed them in the definition of various types of functional dependencies.
(iii) Later in [29] Bosc, Lietard, Pivert defined functional dependencies (called extended functional dependencies) using gradual rules.
(iv) The ideas introduced in [115], see Equations (3.37) *and* (3.38)*, were used in [114] for defining linguistic summaries (also called fuzzy functional dependencies).*

3.3 Comparison of similarity-based generalizations of FD

The semantics of classical FD corresponds to the notion of mathematical function. More precisely: $||A \Rightarrow B||_{\mathcal{D}} = 1$ iff $\{\langle r(A), r(B)\rangle | \, \forall r \in \mathcal{D}\}$ is a function (from $\pi_A(\mathcal{D})$ to $\pi_B(\mathcal{D})$, see (2.43)). In this section we will examine how different approaches correspond to the notion of function. Since the similarity and ranks are employed in the various generalization of Codd's model, the classical definition of function is no longer adequate and we will use the notion of fuzzy function.

The definition of a fuzzy function was provided by S. Gottwald in [63]. In that paper the author introduced a notion of fuzzy uniqueness of a fuzzy mapping F [4] using a formula of first order fuzzy logic. Every fuzzy mapping F has a degree to which it satisfies the uniqueness property U:

$$U(F) = \bigwedge_{x,y,u,v} \Big(((F(x,u) \otimes F(y,v)) \otimes x \equiv y) \rightarrow u \equiv v\Big), \tag{3.51}$$

[4] For the definition of fuzzy mapping as well as for the definition of $\equiv$ see the original paper [63].

where $\rightarrow$ is a Łukasiewicz implication, $\otimes$ stands for a Łukasiewicz t-norm or minimum, i.e. by this definition four different notions of fuzzy uniqueness were given. The set of truth degree was considered as $[0, 1]$. A mapping is called fuzzy function if it is unique to degree 1. Later (3.51) was used by M. Demirci in [55] in the framework of **L**-relation and **L**-equalities. Demirci also used a complete residuated lattice (called integral commutative residuated l-monoid) as a structure of truth degrees.

Definition 6 (Fuzzy function)**.** *Let* **L** *be a residuated lattice, let A and B be crisp sets, and let $\approx_A$ and $\approx_B$ be* **L***-equalities. An* **L***-relation $\rho\colon A \times B \rightarrow L$ (L is a support set of* **L***) is said to be a fuzzy function iff for all $a_1, a_2 \in A$ and $b_1, b_2 \in B$ we have*

$$\rho(a_1, b_1) \otimes \rho(a_2, b_2) \otimes (a_1 \approx_A a_2) \leq (b_1 \approx_B b_2). \tag{3.52}$$

Remark 13. *i) Note that* (3.52) *is only a reformulation of Equation* (3.51) *since in all residuated lattices we have $a \rightarrow b = 1$ iff $a \leq b$.*
ii) Tyagi's definition of GFD [106] was inspired by the condition (3.52)*.*
iii) In [9] the condition (3.52) *corresponds to the notion of compatibility. Relation ρ is called compatible with respect to $\approx_A$ and $\approx_B$ if it satisfies* (3.52)*, where $\approx_A$ and $\approx_B$ are* **L***-equivalencies.*
iv) Another notion of fuzzy function with respect to similarity relation was defined in [67]. The term fuzzy function was understood as a syntactic notion, and the term fuzzy mapping was used as the corresponding semantic one.
5) A partial fuzzy function [55, 72] is used in [9] in the definition of a degree to which a given relation is a fuzzy function.

We will use the idea from [63] (and later from [9]) to define a degree to which a ranked data table corresponds to the notion of fuzzy function given by (3.52).

Definition 7. *Let* **L** *be a complete residuated lattice and $\mathcal{D} : \mathrm{Tupl}(R) \rightarrow L$ be a ranked data table. Let $\approx_i$ be* **L***-similarities on corresponding domains. Let $A, B \subseteq R$ and let the similarity of two tuples on a set of attributes be given by Equation* (3.2)*. The degree to which $\mathcal{D}$ is a fuzzy function with respect to the sets of attributes A and B is defined as:*

$$\mathrm{Fun}(\mathcal{D}, A, B) = \bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} \Big((\mathcal{D}(r_1) \otimes \mathcal{D}(r_2) \otimes (r_1(A) \approx_{\mathcal{D}} r_2(A))) \rightarrow (r_1(B) \approx_{\mathcal{D}} r_2(B)) \Big). \tag{3.53}$$

The verbal description of the previous definition is as follows: The degree to which a relation $\mathcal{D}$ corresponds to a fuzzy function from A to B is a degree to which it is true that for all pairs of tuples from $\mathcal{D}$: if they belong to $\mathcal{D}$ and have similar values on attributes A, then they have similar values on attributes from B.

Remark 14. *(i) In the above definition we only assume that $\approx_i$ are similarities, although in the original works of Demirci and Belohlavek* **L***-equalities and* **L***-equivalences were used, respectively.*
(ii) Note that tuples with zero ranks do not influence the resulting degree of (3.53)*.*

It will not make any difference if we use projection of a ranked data table to $A \cup B$ in the Definition 7 as the following lemma shows:

Lemma 8. *Let* $\mathbf{L}$ *be complete residuated lattice. Given a ranked data table* $\mathcal{D} : \mathrm{Tupl}(R) \to L$ *and a set* $A \subseteq R$*. If the projection of* $\mathcal{D}$ *to* A *is defined as* $\mathcal{D}_A(r) = \sup\{\mathcal{D}(r') \mid r' \in \mathrm{Tupl}(R)$ *with* $r'(A) = r\}$*, then* $\mathrm{Fun}(\mathcal{D}, A, B) = \mathrm{Fun}(\mathcal{D}_{A\cup B}, A, B)$*.*

Proof. Consequence of (2.17) and (2.19). □

We decided not to use the projection in Definition 7 since the definition of projection differs among approaches, although supremum or maximum is usually used. We will now illustrate the above definition on a simple example.

Example 2. *Let* $\mathbf{L}$ *be any complete residuated lattice. Let* $R = \{y_1, y_2, y_3\}$ *with* $D_{y_1} = \{a_1, a_2\}$*,* $D_{y_2} = D_{y_3} = \{b_1, b_2\}$*. Let us assume that the* $\mathbf{L}$*-similarity relation* $\approx_2$ *coincides with* $\approx_3$ *and that* $\approx_1, \approx_2$ *and the RDT* $\mathcal{D}$ *are given as follows:*

$\approx_1$	a_1	a_2
a_1	1	μ_1
a_2	μ_1	1

$\approx_2$	b_1	b_2
b_1	1	μ_2
b_2	μ_2	1

$\mathcal{D}$	y_1	y_2	y_3
λ_1	a_1	b_1	b_1
λ_2	a_2	b_2	b_1
λ_3	a_1	b_1	b_2

The degree to which the relation $\mathcal{D}$ *is a fuzzy function w.r.t.* $\{y_1\}$ *and* $\{y_2\}$ *is computed as follows:*

$$
\begin{aligned}
&\mathrm{Fun}(\mathcal{D}, \{y_1\}, \{y_2\}) = \\
&((\lambda_1 \otimes \lambda_2 \otimes \mu_1) \to \mu_2) \wedge ((\lambda_1 \otimes \lambda_3 \otimes 1) \to 1) \wedge ((\lambda_2 \otimes \lambda_3 \otimes \mu_1) \to \mu_2) = \\
&((\lambda_1 \otimes \lambda_2 \otimes \mu_1) \to \mu_2) \wedge 1 \wedge ((\lambda_2 \otimes \lambda_3 \otimes \mu_1) \to \mu_2) = \\
&((\lambda_1 \otimes \lambda_2 \otimes \mu_1) \to \mu_2) \wedge ((\lambda_2 \otimes \lambda_3 \otimes \mu_1) \to \mu_2) = \\
&((\lambda_1 \otimes \lambda_2 \otimes \mu_1) \vee (\lambda_2 \otimes \lambda_3 \otimes \mu_1)) \to \mu_2 = \\
&((\lambda_1 \vee \lambda_3) \otimes \lambda_2 \otimes \mu_1) \to \mu_2 = \mathrm{Fun}(\mathcal{D}_{\{y_1,y_2\}}, \{y_1\}, \{y_2\}).
\end{aligned}
$$

The last equality holds iff the projection is defined using supremum.

Definition 7 gives us the degree to which a relation (data table $\mathcal{D}$) captures the notion of fuzzy function from A to B. Now we will look at the correspondence of Definition 7 with the degree to which a GFD is true in $\mathcal{D}$ for various definitions of GFD.

The following criterion will give us the degree to which: "For all relations $\mathcal{D}$: If a GFD $A \Rightarrow B$ is satisfied in relation $\mathcal{D}$, then $\mathcal{D}$ corresponds to the fuzzy function from A to B."

$$\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \big(||A \Rightarrow B||_{\mathcal{D}} \to \mathrm{Fun}(\mathcal{D}, A, B)\big). \tag{3.54}$$

Similarly, the next criterion will give us a degree to which: "For all relations $\mathcal{D}$: If $\mathcal{D}$ corresponds to the fuzzy function from A to B, then a GFD $A \Rightarrow B$ is satisfied by $\mathcal{D}$."

$$\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \big(\mathrm{Fun}(\mathcal{D}, A, B) \to ||A \Rightarrow B||_{\mathcal{D}}\big). \tag{3.55}$$

Finally, combining (3.54) and (3.55) we will obtain the degree to which a particular definition of GFD corresponds to the fuzzy function,

$$\begin{aligned}\mathbb{E}(\mathrm{Fun}, A \Rightarrow B) &= \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \big(\mathrm{Fun}(\mathcal{D}, A, B) \leftrightarrow ||A \Rightarrow B||_{\mathcal{D}}\big) \\ &= \mathbb{S}(A \Rightarrow B, \mathrm{Fun}) \wedge \mathbb{S}(\mathrm{Fun}, A \Rightarrow B). \end{aligned} \tag{3.56}$$

The infimum is going over all RDTs on $\mathrm{Tupl}(R)$, where ranks are taken from complete residuated lattice with support L. Note that nonranked data tables are special case of ranked ones with ranks coming from $\{0, 1\}$. Moreover, we would like to remark that the $\to$ (and $\leftrightarrow$) are the operations from the residuated lattice used in the definition of GFD. As a consequence, the concrete form of the criterion (3.56) may vary.

Note that the semantics of GFDs is usually given as follows: similar values on attributes from A correspond to similar values on attributes from B. The semantics of (3.53) is almost the same. The difference is that ranks have impact on the resulting degree. The definition of fuzzy function given by (3.52) is widely accepted among researchers and thus the idea that lower ranks should have lower influence on the validity seems to be appropriate, and we think it is very natural. Contrary to this stands the fact that a lot of GFD definitions do not depend on the ranks at all.

In the rest of this section we will use $\mathbb{E}(\mathrm{Fun}, A \Rightarrow B)$ as a criterion to measure the degree in which a given GFD definition preserves the notion of the fuzzy function. We will compute the criterion (3.56) using (3.54) and (3.55). We have selected significant approaches presented in Section 3 for comparison. Approaches that are similar to the selected ones are not mentioned explicitly, for example the result obtained for Raju and Majumdar's definition is applicable to all their followers.

The following lemma simplifies the computation of (3.54) and (3.55) for the cases where the validity of GFD remains bivalent. We will write $\mathcal{D} \vDash A \Rightarrow B$ and $\mathcal{D} \nvDash A \Rightarrow B$ to denote $||A \Rightarrow B||_{\mathcal{D}} = 1$ and $||A \Rightarrow B||_{\mathcal{D}} = 0$, respectively. The set of all relations that satisfy a given GFD $A \Rightarrow B$ will be denoted as $\mathrm{Mod}(\{A \Rightarrow B\})$ or simply $\mathrm{Mod}(A, B)$.

Lemma 9. *Let* $\mathbf{L}$ *be a complete residuated lattice, let* A *and* B *be sets of attributes* $A, B \subseteq R$*. If the validity of a GFD* $A \Rightarrow B$ *is bivalent then*

$$\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = \bigwedge_{\mathcal{D}\in\mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B), \tag{3.57}$$

$$\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \Big(\bigvee_{\mathcal{D}\notin\mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B)\Big) \to 0. \tag{3.58}$$

Proof. (3.57): Using (2.9), (2.8) and $1 \wedge a = a$ we have:

$$\begin{aligned}\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) &= \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \big(||A \Rightarrow B||_{\mathcal{D}} \to \mathrm{Fun}(\mathcal{D}, A, B)\big) = \\ &= \bigwedge_{\mathcal{D}\in\mathrm{Mod}(A,B)} \big(1 \to \mathrm{Fun}(\mathcal{D}, A, B)\big) \wedge \bigwedge_{\mathcal{D}\notin\mathrm{Mod}(A,B)} \big(0 \to \mathrm{Fun}(\mathcal{D}, A, B)\big) = \end{aligned}$$

$$= \bigwedge_{\mathcal{D}\in \mathrm{Mod}(A,B)} \big(1 \to \mathrm{Fun}(\mathcal{D}, A, B)\big) = \bigwedge_{\mathcal{D}\in \mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B).$$

Equation (3.58) is a consequence of (2.7) and (2.19). Indeed:

$$\begin{aligned}
\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) &= \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \big(\mathrm{Fun}(\mathcal{D}, A, B) \to ||A \Rightarrow B||_{\mathcal{D}}\big) = \\
&= \bigwedge_{\mathcal{D}\in \mathrm{Mod}(A,B)} \big(\mathrm{Fun}(\mathcal{D}, A, B) \to 1\big) \wedge \bigwedge_{\mathcal{D}\notin \mathrm{Mod}(A,B)} \big(\mathrm{Fun}(\mathcal{D}, A, B) \to 0\big) = \\
&= \bigwedge_{\mathcal{D}\notin \mathrm{Mod}(A,B)} \big(\mathrm{Fun}(\mathcal{D}, A, B) \to 0\big) = \big(\bigvee_{\mathcal{D}\notin \mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B)\big) \to 0.
\end{aligned}$$

□

We will now apply the criterion given by Equation (3.56) to all significant approaches.

Theorem 10 (Buckles and Petry)**.** *Let* $\mathbf{L} = [0,1]_{\Pi}$*, let* $A, B \subseteq R$ *be sets of attributes and let the GFD be defined as in* (3.8)*. Assuming* $\beta \in [0,1]$ *is the parameter from* (3.8)*, then* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = \beta$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \beta \to 0$*.*

Proof. First, note that the GFD is defined for nonranked data table and therefore

$$\mathrm{Fun}(\mathcal{D}, A, B) = \bigwedge_{r_1, r_2 \in \mathcal{D}} (r_1(A) \approx_{\mathcal{D}} r_2(A) \to r_1(B) \approx_{\mathcal{D}} r_2(B)). \tag{3.59}$$

Moreover, since $||A \Rightarrow B||_{\mathcal{D}} \in \{0,1\}$ we can use Lemma 9. For proving the first part, observe that if $\mathcal{D} \in \mathrm{Mod}(A,B)$, i.e. $||A \Rightarrow B||_{\mathcal{D}} = 1$, then:

$$\begin{aligned}
&\beta \otimes (r_1(A) \approx r_2(A)) \leq r_1(B) \approx r_2(B) \quad \forall r_1, r_2 \in \mathcal{D}, \\
&\beta \leq (r_1(A) \approx r_2(A) \to r_1(B) \approx r_2(B)) \quad \forall r_1, r_2 \in \mathcal{D}, \\
&\beta \leq \bigwedge_{r_1, r_2 \in \mathcal{D}} (r_1(A) \approx r_2(A) \to r_1(B) \approx r_2(B)).
\end{aligned}$$

The last inequality can be written as $\beta \leq \mathrm{Fun}(\mathcal{D}, A, B)$ for any $\mathcal{D} \in \mathrm{Mod}(A,B)$. As a consequence $\beta \leq \bigwedge_{\mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B)$. Finally, for any β there exists $\mathcal{D} \in \mathrm{Mod}(A,B)$ such that $\mathrm{Fun}(\mathcal{D}, A, B) = \beta$ (take $\mathcal{D}$ with only two tuples $r_1, r_2 \in \mathcal{D}$ such that $r_1(A) \approx r_2(A) = 1$ and $r_1(B) \approx r_2(B) = \beta$). Therefore $\beta = \bigwedge_{\mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B) = \mathbb{S}(A \Rightarrow B, \mathrm{Fun})$. Using our previous observation and Lemma 9 we have:

$$\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \big(\bigvee_{\mathcal{D}\notin \mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B)\big) = \beta \to 0.$$

Note that the proof remains valid for any complete residuated lattice. □

Theorem 11 (Prade and Testemale)**.** *Let* $\mathbf{L}$ *be any complete residuated lattice with universe* $L = [0,1]$*. Let* $A, B \in R$*. Let the GFD be defined by* (3.15) *and let* $\lambda \in [0,1]$ *be the parameter from* (3.15)*. Then* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 0$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \lambda \to 0$*.*

Proof. First note that A, B are single attributes. Moreover, $||A \Rightarrow B||_{\mathcal{D}} \in \{0, 1\}$ and we can apply Lemma 9. For proving $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 0$ it is sufficient to show that $\bigwedge_{\mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B) = 0$. Let us fix four different elements a_1, a_2, b_1, b_2 and consider $\mathcal{M} \subseteq \mathrm{Mod}(A, B)$ being the set of models described as follows:

$\approx_A$	a_1	a_2
a_1	1	α
a_2	α	1

$\approx_B$	b_1	b_2
b_1	1	0
b_2	0	1

$\mathcal{D}$	A	B
1.0	a_1	b_1
1.0	a_2	b_2

where $\alpha \in [0, 1)$ is an arbitrary parameter. It means that the relations from $\mathcal{M}$ differ from each other by the parameter α (by the similarity relation on domain D_A). Then,

$$\bigwedge_{\mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B) \leq \bigwedge_{\mathcal{M}} \mathrm{Fun}(\mathcal{D}, A, B) = \bigwedge_{\alpha \in [0,1)} (\alpha \to 0) = \big(\bigvee_{\alpha \in [0,1)} \alpha \big) \to 0 = 1 \to 0 = 0.$$

The second equality follows from the fact that GFD are defined for nonranked data tables and therefore $\mathrm{Fun}(\mathcal{D}, A, B)$ is given by (3.59). Also note that if $\mathcal{D} \notin \mathrm{Mod}(A, B)$ then there exist tuples r_1, r_2 such that $r_1(A) = r_2(A)$ and $r_1(B) \approx r_2(B) < \lambda$. Therefore

$$\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \Big(\bigvee_{\mathcal{D} \notin \mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B) \Big) \to 0 =$$
$$\Big(\bigvee_{\mathcal{D} \notin \mathrm{Mod}(A,B)} \bigwedge_{r_1, r_2 \in \mathcal{D}} (r_1(A) \approx_{\mathcal{D}} r_2(A) \to r_1(B) \approx_{\mathcal{D}} r_2(B)) \Big) \to 0 = \lambda \to 0.$$

□

Theorem 12 (Raju and Majumdar)**.** *Let* $\mathbf{L}$ *be any complete residuated lattice with universe* $L = [0, 1]$*. Assume* R *is a relational scheme and* $A, B \subseteq R$*. For the GFD given by Equation* (3.19)*,* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 1$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 0$*.*

Proof. First of all, we have to mention that authors' extension involve ranked data tables. As we shall see in this proof, the theorem holds for any residuated lattice built over the unit interval. Observe that if $||A \Rightarrow B||_{\mathcal{D}} = 1$, then $r_1(A) \approx_{\mathcal{D}} r_2(A) \leq r_1(B) \approx_{\mathcal{D}} r_2(B)$ for all $r_1, r_2 \in \mathrm{Tupl}(R)$ with $\mathcal{D}(r_1) > 0$ and $\mathcal{D}(r_2) > 0$. Together with the fact that $\mathcal{D}(r_1) \otimes \mathcal{D}(r_2) \otimes r_1(A) \approx_{\mathcal{D}} r_2(A) \leq r_1(A) \approx_{\mathcal{D}} r_2(A)$, for all r_1, r_2 and any t-norm, we obtain $\mathrm{Fun}(\mathcal{D}, A, B) = 1$ for any $\mathcal{D} \in \mathrm{Mod}(A, B)$. The proof of the first equality is completed by applying Lemma 9.

For proving the second equality it is sufficient to find a ranked data table $\mathcal{D}$ such that $\mathcal{D} \notin \mathrm{Mod}(A, B)$ and $\mathrm{Fun}(\mathcal{D}, A, B) = 1$. Such an RDT is easy to find: consider for example $\mathcal{D}$ with only two tuples r_1, r_2 such that $\mathcal{D}(r_1) = \mathcal{D}(r_2) = 0.2$, $r_1(A) \approx_{\mathcal{D}} r_2(A) = 1$ and $r_1(B) \approx_{\mathcal{D}} r_2(B) = 0.9$. □

Theorem 13 (Chen et al.)**.** *Let* $\mathbf{L} = [0, 1]_G$*,* $A, B \subseteq R$ *be sets of attributes,* $\theta \in [0, 1]$ *and let the GFD be defined as in* (3.24)*. Then* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 1$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 0$*.*

Proof. Note, that if $||A \Rightarrow B||_{\mathcal{D}} = \theta$, then $\mathrm{Fun}(\mathcal{D}, A, B)$ will be at least θ. As a consequence: $||A \Rightarrow B||_{\mathcal{D}} \to \mathrm{Fun}(\mathcal{D}, A, B) = 1$ for any $\mathcal{D}$.

To prove $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 0$ let us consider the set

$$\mathcal{M} = \{\mathcal{D} \text{ over } R = \{A, B\} | \mathcal{D} \nvDash A \Rightarrow B, |\mathcal{D}| = 2 \text{ and } r_1(A) = r_2(A), r_1, r_2 \in \mathcal{D}\}.$$

Note that for each $\mathcal{D} \in \mathcal{M}$ we have $r_1(A) = r_2(A)$ and $r_1(B) \neq r_2(B)$. As a consequence:

$$\begin{aligned}
&\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to\{0,1\}} \big(\mathrm{Fun}(\mathcal{D}, A, B) \to ||A \Rightarrow B||_{\mathcal{D}}\big) \leq \\
&\bigwedge_{\mathcal{D}\in\mathcal{M}} (\mathrm{Fun}(\mathcal{D}, A, B) \to 0) = \\
&\Big(\bigvee_{\mathcal{D}\in\mathcal{M}} (r_1(A) \approx_{\mathcal{D}} r_2(A) \to r_1(B) \approx_{\mathcal{D}} r_2(B)) \Big) \to 0 = \\
&\Big(\bigvee_{\mathcal{D}\in\mathcal{M}} (1 \to r_1(B) \approx_{\mathcal{D}} r_2(B)) \Big) \to 0 = \\
&\Big(\bigvee_{\mathcal{D}\in\mathcal{M}} r_1(B) \approx_{\mathcal{D}} r_2(B) \Big) \to 0 = 1 \to 0 = 0.
\end{aligned}$$

□

Theorem 14 (Bhuniya and Niyogi)**.** *Let* $\mathbf{L} = [0, 1]_L$*,* $A, B \subseteq R$*. For GFD given in* (3.31) *we have* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = \beta$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 0$*.*

Proof. The first result can be proved by following the same arguments as in the proof of Theorem 10. The second result is a consequence of the fact that ranks are not involved in the definition of GFD, see the proof of Theorem 12. □

Theorem 15 (Cubero et al.)**.** *Let* $\mathbf{L}$ *be any complete residuated lattice with universe* $L = [0, 1]$*. For the GFD given by Equation* (3.33) *and for fixed thresholds* c_y*,* $y \in R$*:*

$$\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = \big(\bigvee_{y\in A} c_y \to 0 \big) \wedge \bigwedge_{y\in B} c_y, \tag{3.60}$$

$$\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \big(\bigwedge_{y\in A} c_y \to \bigvee_{y\in B} c_y \big) \to 0. \tag{3.61}$$

Proof. Since the degree to which GFD is true remains bivalent, we can again apply Lemma 9 and compute only $\bigwedge_{\mathrm{Mod}(A,B)} \mathrm{Fun}(\mathcal{D}, A, B)$. First of all notice that $\mathcal{D} \in \mathrm{Mod}(A, B)$ if for all pair of tuples either $r_1(A) \approx r_2(A) < \alpha$ or $r_1(B) \approx r_2(B) \geq \beta$. Since $\alpha = (c_y)_{y\in A}$, $\beta = (c_y)_{y\in B}$ are vectors, $r_1(A) \approx r_2(A) < \alpha$ means there exists $y \in A$ such that $r_1(y) \approx_y r_2(y) < c_y$ and $r_1(B) \approx r_2(B) \geq \beta$ means that for all $y \in B$: $r_1(y) \approx_y r_2(y) \geq c_y$. Now we will look at these two cases separately. First, using (2.18) and isotony of $\to$ in the second argument, we have:

$$\bigwedge_{\mathrm{Mod}(A,B)} \bigwedge_{\substack{r_1,r_2\in\mathcal{D} \\ r_1(A)\approx r_2(A)<\alpha}} \big(r_1(A) \approx r_2(A) \to r_1(B) \approx r_2(B)\big) =$$

$$\bigwedge_{\mathrm{Mod}(A,B)} \bigwedge_{\substack{r_1,r_2\in\mathcal{D}\\ r_1(A)\approx r_2(A)<\alpha}} \big(r_1(A)\approx r_2(A)\to 0\big) =$$

$$\Big(\bigvee_{\mathrm{Mod}(A,B)} \bigvee_{\substack{r_1,r_2\in\mathcal{D}\\ r_1(A)\approx r_2(A)<\alpha}} \big(r_1(A)\approx r_2(A)\big)\Big)\to 0 = \bigvee_{y\in A} c_y \to 0.$$

The last equality follows from the fact that $r_1(A)\approx r_2(A) = \bigwedge_{y\in A} r_1(y)\approx_y r_2(y)$. The second case follows from antitony of $\to$ in the first argument and (2.9):

$$\bigwedge_{\mathrm{Mod}(A,B)} \bigwedge_{\substack{r_1,r_2\in\mathcal{D}\\ r_1(B)\approx r_2(B)\geq\beta}} \big(r_1(A)\approx r_2(A)\to r_1(B)\approx r_2(B)\big) =$$

$$\bigwedge_{\mathrm{Mod}(A,B)} \bigwedge_{\substack{r_1,r_2\in\mathcal{D}\\ r_1(B)\approx r_2(B)\geq\beta}} \big(1\to r_1(B)\approx r_2(B)\big) =$$

$$\bigwedge_{\mathrm{Mod}(A,B)} \bigwedge_{\substack{r_1,r_2\in\mathcal{D}\\ r_1(B)\approx r_2(B)\geq\beta}} \big(r_1(B)\approx r_2(B)\big) = \bigwedge_{y\in B} c_y,$$

finishing the proof of (3.60). The equation (3.61) follows from Lemma (9), antitony of residuum in the first argument and isotony in the second. □

Theorem 16 (Tyagi et al.)**.** *Let* $\mathbf{L} = [0,1]_G$*,* $A, B \subseteq R$ *be sets of attributes and let the GFD be defined as in* (3.43)*. Then* $\mathbb{S}(A\Rightarrow B, \mathrm{Fun}) = 1$ *and* $\mathbb{S}(\mathrm{Fun}, A\Rightarrow B) = 0$*.*

Proof. First of all, the validity of the GFD given by (3.43) is bivalent and therefore we can use Lemma 9 again. The equality $\mathbb{S}(A\Rightarrow B, \mathrm{Fun}) = 1$ follows from the fact that $a\otimes b = a\wedge b$ for all $a, b\in[0,1]_G$ and from (2.4).

The second equality follows from Lemma 9 and the fact that $1\to 0 = 0$. The result is maybe surprising, since Tyagi et at. use the definition of fuzzy function for their GFD. Nevertheless, the validity of their GFD remains bivalent. □

Theorem 17 (Kiss)**.** *Let* $\mathbf{L} = [0,1]_L$*,* $A, B \subseteq R$ *be sets of attributes and let the GFD be defined as in* (3.47)*. Then* $\mathbb{S}(A\Rightarrow B, \mathrm{Fun}) = 1$ *and* $\mathbb{S}(\mathrm{Fun}, A\Rightarrow B) = 0.5$*.*

Proof. The first result follows from (2.12) and from antitony of $\to$ in the first argument.

For proving the second equality we will use (2.20) and $a\to b\leq(b\to c)\to(a\to c)$, which holds in every residuated lattice. Therefore

$$\mathbb{S}(\mathrm{Fun}, A\Rightarrow B) = \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \big(\mathrm{Fun}(\mathcal{D}, A, B)\to ||A\Rightarrow B||_{\mathcal{D}}\big) \geq$$

$$= \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \bigwedge_{r_1,r_2\in\mathcal{D}} \big((D(r_1)\wedge D(r_2)\wedge r_1(A) = r_2(A))\to (D(r_1)\otimes D(r_2)\otimes r_1(A) = r_2(A))\big)$$

$$= \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \bigwedge_{r_1,r_2\in\mathcal{D}} \big((D(r_1)\wedge D(r_2))\to (D(r_1)\otimes D(r_2))\big).$$

Since the $\to$ and $\otimes$ are Łukasiewicz operations we get: $0.5\leq\mathbb{S}(\mathrm{Fun}, A\Rightarrow B)$. It is easy to find relation $\mathcal{D}$ such that $\mathrm{Fun}(\mathcal{D}, A, B)\to||A\Rightarrow B||_{\mathcal{D}} = 0.5$. □

Theorem 18 (Ben Yahia et al.)**.** *Let R be a relational scheme and $A, B \subseteq R$. For GFD given by* (3.37) *and* (3.41)*: $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 1$ and $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 0$.*

Proof. The first result is a consequence of the antitony of $\rightarrow$ in the first argument, more precisely for any $\mathcal{D}$ and any $r_1, r_2 \in \mathcal{D}$ we have:

$$r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B) \leq (\mathcal{D}(r_1) \otimes \mathcal{D}(r_2) \otimes r_1(A) \approx r_2(A)) \rightarrow r_1(B) \approx r_2(B)$$

and thus for any $\mathcal{D}$ we have $||A \Rightarrow B||_{\mathcal{D}} \leq \mathrm{Fun}(\mathcal{D}, A, B)$.

The second result is again a consequence of the fact that the rank is not involved in the definition of GFD and thus it is easy to find a relation $\mathcal{D}$ such that $||A \Rightarrow B||_{\mathcal{D}} = 0$ and $\mathrm{Fun}(A, B) = 1$. □

Theorem 19 (Bosc et al.)**.** *Let R be a relational scheme and $A, B \subseteq R$. For GFD given by* (3.41)*: $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 1$ and $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 1$.*

Proof. Since authors proposed the definition of GFD for nonranked data tables, we have for all $\mathcal{D}$: $\mathrm{Fun}(\mathcal{D}, A, B) = ||A \Rightarrow B||_{\mathcal{D}}$. □

Theorem 20 (Belohlavek and Vychodil)**.** *Let R be a relational scheme and $A, B \subseteq R$. For the GFD defined by Equation* (2.62) *we have: $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 1$ and $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 1$.*

Proof. According to (2.63) ranks are involved in the definition of similarity itself and not in definition of GFD, therefore it seems somehow inadequate to apply the criteria given by (3.56). Fortunately, the authors have proved in [24] that for each ranked data table $\mathcal{D}$ there exists a nonranked $\mathcal{D}'$ (ranks come from $\{0, 1\}$) such that $||A \Rightarrow B||_{\mathcal{D}} = ||A \Rightarrow B||_{\mathcal{D}'}$. Therefore for hedge being identity we obtain: $\mathrm{Fun}(\mathcal{D}, A, B) = ||A \Rightarrow B||_{\mathcal{D}'} = ||A \Rightarrow B||_{\mathcal{D}}$. □

Theorem 21 (Cordero et al. case)**.** *Let R be a relational scheme and $A, B \subseteq R$. If the GFD is defined by Equation* (3.49)*, then $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 1$ and $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 1$.*

Proof. Consequence of Lemma 5 and Theorem 20. □

We have seen that although the definition of fuzzy function is natural and widely accepted, many approaches to GFD failed to satisfy the criterion given by Equation (3.56). One reason is that the validity of GFD usually remains crisp. Another reason is the fact that although many of the GFDs are defined for some rank-aware model, the ranks (usually) do not influence the validity of such dependency.

3.4 Conclusions

The main aim of this chapter was to compare generalizations of FD in which similarity relations replace the classical equality. We have established a criterion which makes the comparison of various GFD easier and more objective. The criterion given by Equation (3.56) gives us a degree to which a particular GFD corresponds to the fuzzy function.

From various notions of GFD we have selected 12 of them which have had significant impact on other authors and have introduced interesting and new approach to GFD: Buckles and Petry (1983), Prade and Testemale (1984), Raju and Majumdar (1988), Chen (1991), Bhuniya and Niyogi (1993), Cubero et al. (1994), Ben Yahia et al. (1999), Bosc et al. (1999), Tyagi et al. (2005), Kiss (1991), Belohavek and Vychodil (2006) and Cordero et al. (2011). The summary can be found in Table 3.2. We want to emphasize several points:

Authors/approach	GFD	*[Imp]*	*[TrGFD]*	*[Rank]*	$\mathbb{E}(\text{Fun}, A \Rightarrow B)$
Buckles and Petry [31]	(3.6)	R-G imp.	$\{0, 1\}$	No	$\beta \wedge (\beta \to 0)$
Prade and Testemale [93]	(3.15)	R-G imp.	$\{0, 1\}$	No	0
Raju and Majumdar [95]	(3.19)	R-G imp.	$\{0, 1\}$	Yes	0
Chen et al. [61]	(3.24)	Classical or Gödel	$\{0, 1\}$	No	0
Bhuniya and Niyogi [26]	(3.27)	R-G imp.	$\{0, 1\}$	Yes	0
Cubero et al. [48]	(3.33)	R-G imp.	$\{0, 1\}$	No	(3.60)$\wedge$(3.61)
Tyagi et al. [106]	(3.43)	R-G imp.	$\{0, 1\}$	Yes	0
Kiss [71]	(3.45)	Łukasiewicz	$[0, 1]$	Yes	0.5
Ben Yahia et al. [115]	(3.37)	Łukasiewicz	$[0] \cup [\theta, 1]$	Yes	0
Bosc, Pivert and Ughetto [30]	(3.41)	Residuum	$[0, 1]$	No	1
Belohavek and Vychodil [21]	(3.37)	Residuum	Complete residuated lattice	Yes	1
Cordero et al. [45]	(3.49)	Residuum	$[0, 1]$	Yes	1

Table 3.2: Review of similarity-based functional dependencies.
In the *[Imp]* column the implication used in definition of a GFD is highlighted. The choice of the implication influences the degree to which a GFD is true, column *[TrGFD]*. The column *[Rank]* indicates if a GFD is defined for data table with ranks. In the last column the degree to which a GFD corresponds to fuzzy function is presented.

1. As it is shown in Table 3.2, many approaches reduce the new (generalized) concept of FD to a bivalent one. This is usually done by introducing some extra parameter and by letting the GFD to be satisfied when some criterion exceeds the parameter. Otherwise the GFD is not satisfied. We strongly believe that a proper approach (from the logical point of view) should consider a richer framework.

2. In some cases the interpretation of a rank is not very clear.

3. The ranks are not usually involved in the definition of GFD. This fact yields to odd behavior: tuples with very low ranks may caused the GFD to be satisfied to low

degree (even 0). Note that our criterion (Equation (3.56)) is able to capture this kind of behavior.

4. As we have seen, in some cases, the generalizations of relational model and functional dependencies are not treated consistently. For example in the Buckles-Petry's model the GFD is defined for nonranked data tables. Nevertheless, the ranks may appear after executing a query. Does the GFD remain the same for ranked data table or is the definition not applicable?

5. In many cases, the authors put their effort only to GFD. As we have already mentioned, the GFD from [30] were not developed any further—even Armstrong-like axioms were not introduced.

None of these problems appears in approaches which are built on fuzzy logic in narrow sense [21, 45]. Among conceptual clarity, the connection to fuzzy logic in narrow sense enables us to generalize many concepts form the original Codd's relational model (which is connected to the first order logic). In the next chapter we will provide a generalization of derivation graphs [81] which can be seen as an alternative prove system.

As a future work, we want to focus on a deeper comparison of relational languages for similarity-based queries: although relational algebra is usually defined, no completeness with respect to domain calculus is presented. And if so, the domain calculus is usually based on classical predicate logic. As far as we know the completeness of relational algebra (which involves similarity-based queries) with respect to domain calculus based on fuzzy predicate logic was provided only in [20], see [23] for current results.

We have provided only one criterion for comparing GFDs which takes similarities into account. As a part of the future work we want to provide criteria which will reflect sensitivity issues in the following sense:

1. Having similar data tables $\mathcal{D}_1$, $\mathcal{D}_2$ (e.g. results of similarity-based queries in two database instances), can we say anything about the validity of a GFD in $\mathcal{D}_1$ and $\mathcal{D}_2$? Is the definition of GFD insensitive to small changes in data and (or) ranks or is it not?

2. For similarity-based relational algebras: If the similarity-based query is evaluated on similar data tables, will the results be similar as well?

One may assume that a similarity-based model is robust if a small change of input data and/or domains similarities and/or ranks leads only to a small change in the query result and to a small change in validity of GFD. Although this requirement is quite natural, it is not obvious if it is satisfied (and under which conditions) by the currently available approaches. In Chapters 5 and 6 we will study the sensitivity issues for one particular model: ranked data tables over domains with similarities from Section 2.3.

Chapter 4

Derivation digraphs for graded if-then rules

Functional dependencies (2.44) are rules of the form $A \Rightarrow B$, where A, B are sets of attributes, and play an important role in relational database systems [41, 82]. From the point of view of syntax, functional dependencies are the same formulas (if-then rules) as attribute implications in formal concept analysis (FCA), see [62], but their interpretation is different. Attribute implications are interpreted in formal context, which is a triple X, Y, I, where X is a set of objects, Y is a set of attributes, and I is a binary relation $I \subseteq X \times Y$ indicating which object has which attribute. The basic meaning of $A \Rightarrow B$ in FCA is that if an object has all the attributes from A, then it has all the attributes from B. An interesting property is that both the different interpretations of the if-then rules yield the same notion of semantic entailment. As a result, one can use a single inference system for reasoning with both attribute implications and functional dependencies. The best known inference system has been proposed by Armstrong [1] and can be simplified to a system of two rules [69]. An interesting alternative graph-based approach that is also aimed at possible automated proving has been proposed by Maier in [81], see also [82] for an extensive description and its application for theorem proving.

In this chapter we present a graph-based method of reasoning with graded if-then rules, by which we mean rules of the form $A \Rightarrow B$, where A, B are fuzzy sets of attributes. Rules of this form describe dependencies between attributes in ordinal and similarity-based data and have two basic interpretations:

1. Similarity-based functional dependencies, see Section 2.3.3, which are interpreted in ranked data tables.

2. Attribute implications (AIs) in formal concept analysis with grades [13], where objects are allowed to have attributes (features) to degrees, i.e. $I \subseteq L^{X \times Y}$. Given $M \in L^Y$ (**L**-set of attributes) the degree to which $A \Rightarrow B$ is true in M is degree to which: "if it is very true, that the object has all attributes from A, then it has also all attributes from B." Formally: $||A \Rightarrow B||_M = S(A, M)^* \rightarrow S(B, M)$.

Even in the graded version, the notion of semantic entailment for SBFDs coincide with the notion of semantic entailment for AIs in FCA with grades in the following sense: The degree to which a graded if-then rule $A \Rightarrow B$ follows from a theory (**L**-set of graded if-then rules) is the same under both interpretations [13]. As a consequence, on may use single Armstrong-like axiomatization, for example the rules (Ax), (Cut) and (Mul) from Section 2.3.3.

Looking for a graph-based inference system for graded if-then rules is interesting from several viewpoints. First, the notion of semantic entailment of the rules we consider is graded, i.e., the entailment expresses a degree to which a rule follows from other rules. It is therefore interesting to find a graph-based inference system that is able to infer rules from other ones including the entailment degrees. Second, there is an Armstrong-like axiomatization of the semantic entailment for the graded rules (see Section 2.3.3, or the original papers [15, 13]), i.e., one might be interested in finding a corresponding graph-based inference method. Third, the Armstrong-like proofs can be formalized to form particular sequences (so-called MRAP-sequences, see [17]). It is therefore interesting to observe whether the graph-based proofs can be constructed according to the normalized proofs and *vice versa.*

In what follows the graded if-then rules will be called fuzzy attribute implications (FAIs). We will first introduce derivation digraphs as particular labeled acyclic digraphs constructed from an input theory (collections of FAIs).

4.1 Derivation acyclic digraphs for FAIs

We now introduce derivation digraphs as particular acyclic digraphs where vertices are labeled by attributes from R and degrees from **L**. The arcs of the digraphs will correspond to FAIs from an input theory and indicate which formulas from the theory are used in the process of inference. In what follows, **L** is a complete residuated lattice. In order to denote that * is a hedge on **L**, we write **L***.

Definition 22 (T-based **L***-derivation DAG)**.** Let T be a set of FAIs over R.

1. Any $\mathbf{D} = \langle V, \emptyset \rangle$ such that $\emptyset \neq V \subseteq R \times L$ and for every $y \in R$ there is at most one $a \in L$ such that $\langle y, a \rangle \in V$, is a T-based **L***-derivation DAG;
2. If $\mathbf{D} = \langle V, A \rangle$ is a T-based **L***-derivation DAG and there are $E \Rightarrow F \in T$, attribute $y \in R$, and vertices $\langle y_1, a_1 \rangle \in V, \ldots, \langle y_k, a_k \rangle \in V$ such that for

$$s_0 = \bigwedge \{E(y) \to 0 \,|\, y \in R \text{ and } y \notin \{y_1, \ldots, y_k\}\}, \tag{4.1}$$

$$s_1 = \bigwedge \{E(y_i) \to a_i \,|\, i = 1, \ldots, k\}, \tag{4.2}$$

$$m = \bigvee \{a \in L \,|\, \langle y, a \rangle \in V\}, \tag{4.3}$$

$$d = \big((s_0 \wedge s_1)^* \otimes F(y)\big) \vee m, \tag{4.4}$$

we have $d > m$, then $\mathbf{D}' = \langle V', A' \rangle$, where

$$V' = V \cup \{\langle y, d \rangle\}, \tag{4.5}$$

$$A' = A \cup \{\langle\langle y_i, a_i\rangle, \langle y, d\rangle\rangle \mid i = 1, \ldots, k\}, \tag{4.6}$$

is a T-based $\mathbf{L}^*$-derivation DAG.

Remark 15. *(i) As one can see, the definition of T-based $\mathbf{L}^*$-derivation DAGs is recursive. The base step says that a set of unconnected vertices is a T-based $\mathbf{L}^*$-derivation DAG if for every $y \in R$ we have $|\{a\,;\, \langle y, a\rangle \in V\}| \leq 1$. Notice that the set V of vertices can be seen as a partial map from R to L. The meaning of the vertices in V is the following: if $\langle y, a\rangle \in V$, we can interpret the fact that the attribute y is assumed valid at least to degree a. Thus, the DAG defined by the base step represents a fact that some attributes are assumed valid to some (nonzero) degrees and we do not make any assumptions about the remaining attributes (not present in the vertices).*

(ii) In the second step, the definition postulates that more complex T-based $\mathbf{L}^*$-derivation DAGs result from simpler ones by adding a vertex and arcs leading from vertices related to antecedents of FAIs from T. In more detail, the idea is that one selects an attribute y which is assumed to be valid to a degree denoted by m, see (4.3)*, and the validity of which can be increased to a strictly higher degree d, see* (4.4)*, by considering a FAI $E \Rightarrow F \in T$ with the following properties: (I) there are vertices $\langle y_1, a_1\rangle, \ldots, \langle y_k, a_k\rangle$ the validities of which are at least $E(y_i)$ for each $i = 1, \ldots, k$; (II) the thresholds prescribed by E for attributes not among those in $y_1, \ldots, y_k$ are zero; (III) d is obtained as a supremum of m (the assumed validity) and the degree to which it is "very true that (I) and (II) hold" and "y is prescribed by F". If the condition (III) holds and $d > m$, the original T-based $\mathbf{L}^*$-derivation DAGs can be extended by vertex labeled by $\langle y, d\rangle$ and arcs going from the selected vertices to the new vertex. Note that the conditions we have just described correspond to expressions* (4.2)*,* (4.1)*, and* (4.4)*, respectively.*

(iii) For better understanding of the second step in Definition 22, consider the case when $\mathbf{L}$ is a two-element Boolean algebra. Suppose we have $\mathbf{D}$ with all vertices of the form $\langle y, 1\rangle$ for all $y \in R'$ where R' is a subset of R. Then, in the second step of Definition 22, $a_1 = \cdots = a_k = 1$ and the step is applied to form $\mathbf{D}'$ whenever the following conditions hold: $s_0 = 1$ (which is true iff E, considered as an ordinary set, consists at most of the attributes $y_1, \ldots, y_k$), $s_1 = 1$ (holds trivially), $m = 0$ iff $\langle y, 1\rangle \notin V$, $d = 1$ iff $y \in F$ (F considered as an ordinary set). It means, T contains $E \Rightarrow F$ such that all attributes appearing in E (to nonzero degrees) are already contained in $\mathbf{D}$, and F has an attribute y that is not contained in $\mathbf{D}$. This can be interpreted so that the antecedent of $E \Rightarrow F$ is "proved valid by $\mathbf{D}$" and thus, we may construct a new DAG $\mathbf{D}'$ which in addition "proves that y is valid". This particular deduction step corresponds to computing closures of attribute sets for ordinary functional dependencies [6, 82]. The Definition 22 can be seen as graded extension of this procedure.

If $\mathbf{D}$ is a T-based $\mathbf{L}^*$-derivation DAG, we put

$$\mathbf{D}(y) = \bigvee\{a \in L \mid \langle y, a\rangle \in V\}, \tag{4.7}$$

and call $\mathbf{D}(y)$ the *yield of* $\mathbf{D}$ *on* y. Clearly, the yield of $\mathbf{D}$ corresponds to (4.3), i.e., we can interpret it as the degree to which y is assumed to be valid according to $\mathbf{D}$. Moreover, $\langle y, a\rangle \in V$ is called an *initial vertex* of $\mathbf{D}$ if $\langle y, a\rangle$ has no incoming arcs (i.e., no arc in $\mathbf{D}$ enters $\langle y, a\rangle$).

Notice that for each $y \in R$ such that $\mathbf{D}(y) > 0$ there is $\langle y, a\rangle \in V$ such that $a = \mathbf{D}(y)$. This is a consequence of Definition 22. Furthermore, it follows that for any $y \in R$, the set

$$L_y = \{a \in L \mid \langle y, a\rangle \in V\} \tag{4.8}$$

has a greatest element provided that $L_y \neq \emptyset$. Another direct consequence of Definition 22 is that L_y is either empty or it is a finite subchain (if equipped with the restriction of $\leq$ to L_y) of the lattice part of $\mathbf{L}$. The latter observation is of course trivial if $\mathbf{L}$ is a chain but it pertains to all complete residuated lattices taken for $\mathbf{L}$. We make use of these observations later in the proofs.

The following notion introduces derivation digraphs related to FAIs:

Definition 23 (T-based $\mathbf{L}^*$-derivation DAG for $E \Rightarrow F$)**.** Let $\mathbf{D} = \langle V, A\rangle$ be a T-based $\mathbf{L}^*$-derivation DAG. Then $\mathbf{D}$ is called a T-based $\mathbf{L}^*$-derivation DAG for $E \Rightarrow F$ if the following conditions are all satisfied:

1. $\mathbf{D}(y) \geq F(y)$ for all $y \in R$;
2. if $E \neq \emptyset$ then the set of initial vertices of $\mathbf{D}$ is

$$\{\langle y, E(y)\rangle \mid y \in R \text{ and } E(y) > 0\}; \tag{4.9}$$

3. if $E = \emptyset$, then the set of initial vertices of $\mathbf{D}$ is $\{\langle y^\sharp, 0\rangle\}$,

where $y^\sharp \in R$ is a designated attribute.

By the designated attribute in the previous definition we mean a fixed attribute that has been selected from R (no particular role or intended interpretation of the attribute is assumed). In theory, we could have defined the set of initial vertices for any E as $\{\langle y, E(y)\rangle \mid y \in R\}$ but this would introduce extraneous vertices into the DAG and that can be seen as an undesirable feature especially from the computational point of view (imagine situation when R is large compared to the number of attributes which belong to E to nonzero degrees). Therefore, we have distinguished the cases for $E = \emptyset$ and $E \neq \emptyset$, because we want to have a minimum set of initial vertices so that $\mathbf{D}$ remains a DAG.

Example 3. *In this example, we utilize the residuated lattice with* $L = [0, 1]$ *given by the Lukasiewicz operations together with hedge* * *defined as follows: For each* $a \in L$ *we put*

$$a^* = \begin{cases} 1, & \text{for } a = 1, \\ 0.6, & \text{for } 0.6 \leq a \leq 0.9, \\ 0.2, & \text{for } 0.2 \leq a \leq 0.5, \\ 0, & \text{for } 0 \leq a \leq 0.1. \end{cases}$$

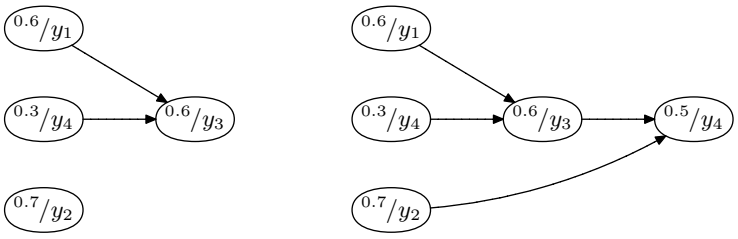


Figure 4.1: Construction of T-based $\mathbf{L}^*$-derivation DAG.

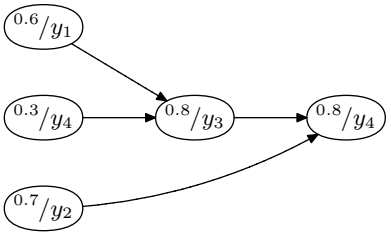


Figure 4.2: T-based $\mathbf{L}^*$-derivation DAG for * being identity.

Figure 4.1 depicts a single step of the process of construction of a T-based $\mathbf{L}^*$-derivation DAG for FAI $\{^{0.6}/y_1, {}^{0.7}/y_2, {}^{0.3}/y_4\} \Rightarrow \{^{0.5}/y_4\}$, where T is the following set of FAIs:

$$T = \{\{^{0.7}/y_1, {}^{0.5}/y_4\} \Rightarrow \{^{0.7}/y_2, {}^{1}/y_3\}, \{^{0.7}/y_3\} \Rightarrow \{^{0.8}/y_5\},$$
$$\{^{0.7}/y_2, {}^{0.9}/y_3\} \Rightarrow \{^{0.9}/y_4\}\}.$$

The DAG on the right-hand side of Figure 4.1 results from the DAG on the left-hand side by adding vertex $\langle y_4, 0.5\rangle$ and two arcs leading from $\langle y_2, 0.7\rangle$ and $\langle y_3, 0.6\rangle$. In a more detail, the vertex $\langle y_4, 0.5\rangle$ in the right-hand side DAG has been added because the validity of y_4 resulting from some $E \Rightarrow F \in T$ was strictly higher than 0.3. Namely, for $E = \{^{0.7}/y_2, {}^{0.9}/y_3\}$, $F = \{^{0.9}/y_4\}$, we have $m = 0.3$, $(s_0 \wedge s_1)^* \otimes F(y_4) = 0.7^* \otimes 0.9 = 0.6 \otimes 0.9 = 0.5$, see (4.1)–(4.4). If we replace the hedge by identity, then the T-based $\mathbf{L}^*$-derivation DAG for the same fuzzy attribute implication as before looks like the one in Figure 4.2. Note that this time the $\mathbf{L}^*$-derivation DAG is even a DAG for $\{^{0.6}/y_1, {}^{0.7}/y_2, {}^{0.3}/y_4\} \Rightarrow \{^{0.8}/y_4\}$.

4.2 Completeness

We now turn our attention to the completeness by which we mean a characterization of the semantic entailment by existence of $\mathbf{L}^*$-derivation DAGs. We prove the claim by showing that a FAI is provable from a theory T iff it has a T-based $\mathbf{L}^*$-derivation DAG. We now show that T-based $\mathbf{L}^*$-derivation DAGs are in a correspondence with normalized proofs called MRAP-sequences [17].

Recall from [17] that the following three rules can be derived from (Ax), (Cut) and (Mul), which were introduced in Section 2.3.3:

(Ref) infer $A \Rightarrow A$,

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

(Pro) from $A \Rightarrow B \cup C$ infer $A \Rightarrow B$,

for all $A, B, C, D, E \in L^Y$. The rules are called reflexivity, accumulation and projection, respectively. By a derivable rule we mean that for all $A, B, C, D, E \in L^Y$, from the part preceding "infer", we can derive using (Ax), (Mul), and (Cut), the part succeeding "infer".

By an *MRAP-sequence for $A \Rightarrow B$ from T* (see [17]), we mean a sequence of formulas such that it

(a) starts with $A \Rightarrow A$;

(b) continues with FAIs from T;

(c) continues with FAIs which result from using (Mul) on FAIs from (b);

(d) continues with FAIs which result from using (Acc) on FAIs from (a), (b), (c), (d);

(e) ends with a single application of (Pro), on the last FAI in (d);

(f) the FAI which results by (e) is $A \Rightarrow B$.

In [17] the following assertion was proved:

Theorem 24. *Let T be a set of FAIs. Then the following is equivalent:*
1) $||A \Rightarrow B||_T = 1$,
2) $A \Rightarrow B$ is provable from T using (Ax), (Cut), (Mul),
3) there is an MRAP-sequence for $A \Rightarrow B$ from T.

Before presenting the equivalence between 3) from the previous theorem and existence of T-based $\mathbf{L}^*$-derivation DAG, we introduce even a more restrictive notion of a derivation sequence by putting further restriction on (d). Namely, we may require that all formulas appearing in (d) should have A as their antecedents.

Lemma 25. *Any MRAP-sequence for $A \Rightarrow B$ and T can be transformed into an MRAP-sequence for $A \Rightarrow B$ and T such that all formulas appearing in its (d)-part are of the form $A \Rightarrow C$.*

Proof. For any MRAP-sequence for $A \Rightarrow B$ and T, one can use the same argument as in [82, Theorem 4.2, page 55] because (Acc) is just an "ordinary rule" with ordinary sets replaced by $\mathbf{L}$-sets. □

From now on, we tacitly assume that all MRAP-sequences satisfy the additional condition justified by Lemma 25. The following assertions show constructions of MRAP-sequences based on T-based $\mathbf{L}^*$-derivation DAGs and *vice versa.*

Theorem 26. *Let T be a theory. If there is an MRAP-sequence for $A \Rightarrow B$ from T, then there is a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$.*

Proof. Let us assume there is an MRAP-sequence for $A \Rightarrow B$ from T and let $A \Rightarrow X_1, \ldots, A \Rightarrow X_n$ be all FAIs from the sequence that have A as their antecedents. We construct a sequence of T-based $\mathbf{L^*}$-derivation DAGs $\mathbf{D}_i$ $(i = 1, \ldots, n)$ such that $\mathbf{D}_i$ is a T-based $\mathbf{L^*}$-derivation DAG for all $A \Rightarrow X_j$ $(j \leq i)$. In addition, each $\mathbf{D}_{i+1}$ results from $\mathbf{D}_i$ by a series of finitely many applications of the second rule of Definition 22. We distinguish the following cases based on the role of $A \Rightarrow X_i$ in the MRAP-sequence.

Case a) By definition of an MRAP-sequence, $A \Rightarrow X_1$ must be an instance of (Ref), i.e., $X_1 = A$. In that case, we let $\mathbf{D}_1$ be the DAG which consists solely of the initial vertices (and no arcs) corresponding to A.

Case b) Let $E \Rightarrow F \in T$ be the first FAI in the MRAP-sequence such that $E = A$. Since all attributes from A are already present in $\mathbf{D}_1$, we can enlarge $\mathbf{D}_1$ by attributes y for which $F(y) \vee \mathbf{D}_1(y) > \mathbf{D}_1(y)$ according to the rule 2 in Definition 22. Note that $(s_0 \wedge s_1)^* = 1^* = 1$. Doing so we will obtain an $\mathbf{L^*}$-derivation DAG $\mathbf{D}_2$ for $A \Rightarrow F$. We repeat the process for all FAIs in the MRAP-sequence that are from T and have A as their antecedents and form a sequence of DAGs $\mathbf{D}_2, \ldots, \mathbf{D}_j$ for some $j \leq n$.

Case c) Let $c^* \otimes E \Rightarrow c^* \otimes F$ be the first FAI in the MRAP-sequence which results from $E \Rightarrow F \in T$ by (Mul) and $c^* \otimes E = A$. Analogously to the Case b), we consecutively enlarge $\mathbf{D}_j$ by attributes y for which $d > \mathbf{D}_j(y)$, where $d = ((s_0 \wedge s_1)^* \otimes F(y)) \vee \mathbf{D}_j(y)$. Doing so we will obtain a new DAG $\mathbf{D}_{j+1}$ which is also a DAG for $A \Rightarrow c^* \otimes F$. Indeed, we have

$$s_0 \wedge s_1 = \bigwedge_{y \in R} \big(E(y) \to A(y)\big) = \bigwedge_{y \in R} \big(E(y) \to (c^* \otimes E(y))\big) \geq c^*,$$

and so for every y that is used in the construction of $\mathbf{D}_{j+1}$ from $\mathbf{D}_j$, we have $d \geq (s_0 \wedge s_1)^* \otimes F(y) \geq c^* \otimes F(y) = (c^* \otimes F)(y)$ due to monotony and idempotency of *. We repeat the process for the remaining FAIs from the MRAP-sequence which result by (Mul) and have A as their antecedent. We will form a sequence of DAGs $\mathbf{D}_{j+1}, \ldots, \mathbf{D}_k$ for some $k \leq n$.

Case d) Let $A \Rightarrow X_l$ $(l > k)$ results by using (Acc). According to Lemma 25, $A \Rightarrow X_l$ results from some FAIs $A \Rightarrow G \cup C$ and $C \Rightarrow D \cup E$, i.e., $X_l = G \cup C \cup D$. Since $\mathbf{D}_{l-1}$ is already a T-based $\mathbf{L^*}$-derivation DAG for $A \Rightarrow G \cup C$, we have $C(y) \leq \mathbf{D}_{l-1}(y)$ for all $y \in R$ and we may proceed with subcases as follows:

1. If $C \Rightarrow D \cup E$ is from T, we proceed analogously as in the Case b).
2. If $C \Rightarrow D \cup E$ results from application of (Mul) on some FAI from T, we proceed analogously as in the Case c).
3. If $C \Rightarrow D \cup E$ results from application of (Acc), then $C = A$ and $\mathbf{D}_{l-1}$ is a T-based $\mathbf{L^*}$-derivation DAG for $C \Rightarrow D \cup E$ (trivial case).

Thus, $\mathbf{D}_l$ results from $\mathbf{D}_{l-1}$ by consecutive addition of vertices and arcs by one of the preceding subcases. We repeat the process for all FAIs obtained by (Acc) and form a sequence of DAGs $\mathbf{D}_{k+1}, \ldots, \mathbf{D}_{n-1}$.

Case e) The last FAI $A \Rightarrow X_n$ in the MRAP-sequence is obtained using (Pro) on $A \Rightarrow X_{n-1}$. Notice that $\mathbf{D}_{n-1}$ is already a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow X_n$ because we have $X_n(y) \leq X_{n-1}(y)$ for all $y \in R$. Hence, we may let $\mathbf{D}_n = \mathbf{D}_{n-1}$.

$\mathbf{D}_n$ is the desired T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$ since $B = X_n$. □

In the opposite direction, we have the following characterization.

Theorem 27. *Let T be a theory. If there is a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$, then there is an MRAP-sequence for $A \Rightarrow B$ from T.*

Proof. Let $\mathbf{D}$ be a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$. We will create an MRAP-sequence for $A \Rightarrow B$. The proof goes by induction on the complexity of T-based $\mathbf{L}^*$-derivation DAGs. Let $\mathbf{D}_1, \ldots, \mathbf{D}_n$ be a sequence of T-based $\mathbf{L}^*$-derivation DAGs such that $\mathbf{D}_{i+1}$ is constructed from $\mathbf{D}_i$, $i \in \{1, \ldots, n\}$, by the second rule of Definition 22. We will create an MRAP-sequence with $A \Rightarrow B_{\mathbf{D}_i}$ as a subsequence, where $B_{\mathbf{D}_i}$ is a yield of $\mathbf{D}_i$, i.e. $B_{\mathbf{D}_i} \in L^R$ such that $B_{\mathbf{D}_i}(y) = \mathbf{D}_i(y)$ for all $y \in R$.

$\mathbf{D}_1$ consists solely of the initial vertices, and our MRAP-sequence will start with (Ref): $T \vdash A \Rightarrow A$.

Assume that $\mathbf{D}_{i+1}$ results from $\mathbf{D}_i = \langle V_i, A_i \rangle$ by the second rule of Definition 22. Therefore, there are $E \Rightarrow F \in T$, attribute $y \in R$, and vertices $\langle y_1, a_1 \rangle \in V_i, \ldots, \langle y_k, a_k \rangle \in V_i$ such that $\mathbf{D}_{i+1}$ results by adding vertex $\langle y, d \rangle$, where $d = ((s_0 \wedge s_1)^* \otimes F(y)) \vee \mathbf{D}_i(y)$ for s_0 and s_1 given by (4.1) and (4.2), respectively. Therefore, $B_{\mathbf{D}_{i+1}} \subseteq \{^d/_y\} \cup B_{\mathbf{D}_i}$. Consider now $G \subseteq B_{\mathbf{D}_i}$ such that $G(y_i) = a_i$ (for all $i = 1, \ldots, k$), and $G(y') = 0$ for all $y' \in R$ which are not among $y_1, \ldots, y_k$. In order to make G well defined, we have to assume that all $y_1, \ldots, y_k$ are pairwise distinct. We may indeed assume this since for $y_{i_1} = \cdots = y_{i_k}$ we may substitute selected vertices $\langle y_{i_1}, a_{i_1} \rangle, \ldots, \langle y_{i_k}, a_{i_k} \rangle$ by a single vertex $\langle y_{i_1}, \bigwedge_{j=1}^k a_{i_j} \rangle$ where $\bigwedge_{j=1}^k a_{i_j}$ is in fact one of the degrees $a_{i_1}, \ldots, a_{i_k}$ (recall that L_y given by (4.8) is a finite chain and so are its arbitrary nonempty subsets). From induction hypothesis $\mathbf{D}_i$ is a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B_{\mathbf{D}_i}$. Therefore,

$$
\begin{array}{ll}
T \vdash A \Rightarrow B_{\mathbf{D}_i} & \text{induction hypothesis} \\
T \vdash E \Rightarrow F & \text{from } E \Rightarrow F \in T \\
T \vdash S(E,G)^* \otimes E \Rightarrow S(E,G)^* \otimes F & \text{using (Mul)} \\
T \vdash A \Rightarrow \underbrace{(S(E,G)^* \otimes \{^{F(y)}/_y\}) \cup B_{\mathbf{D}_i}}_{\{^{S(E,G)^* \otimes F(y)}/_y\} \cup B_{\mathbf{D}_i}} & \text{using (Acc)}
\end{array}
$$

Now, observe that $s_0 \wedge s_1 = S(E,G)$, i.e., $T \vdash A \Rightarrow \{^d/_y\} \cup B_{\mathbf{D}_i}$, see (4.4). Notice that the application of (Acc) is correct since $S(E,G)^* \otimes E \subseteq G \subseteq B_{\mathbf{D}_i}$ and $S(E,G)^* \otimes \{^{F(y)}/_y\} \subseteq S(E,G)^* \otimes F$. Then, we can transform the resulting sequence into an MRAP-sequence by reordering its formulas: we start with $A \Rightarrow A$, continue with formulas from T, then formulas resulting from previous ones by using (Mul), then formulas resulting by applications of (Acc) which all have A as their antecedents, and ending with the application of (Pro). □

Note that the proofs of the previous assertions do not utilizes any notions from semantics of FAIs, and so they are presented in a purely proof-theoretical way. The following assertion provides the ordinary-style completeness:

Theorem 28. *If* $\mathbf{L}$ *is finite, then* $||A \Rightarrow B||_T = 1$ *iff there is a* T*-based* $\mathbf{L}^*$*-derivation DAG for* $A \Rightarrow B$*.*

Proof. Follows from Theorem 26, Theorem 27 and Theorem 24. □

Furthermore, we can express the graded-style completeness as follows:

Theorem 29. *If* $\mathbf{L}$ *is finite, then* $||A \Rightarrow B||_T$ *is the greatest degree* $a \in L$ *such that there is a* T*-based* $\mathbf{L}^*$*-derivation DAG for* $A \Rightarrow a \otimes B$*.*

Proof. Consequence of Theorem 28 and (2.67). □

In abstract fuzzy logic (also known as Pavelka's fuzzy logic [67, 89, 90, 91]), theories as considered as $\mathbf{L}$-sets of formulas. $T(\varphi)$ is interpreted as a degree to which T prescribes φ valid. Even in this case, we can show that T-based $\mathbf{L}^*$-derivation DAGs are capable of describing degrees of semantic entailment as it is shown by the following theorem.

Theorem 30. *If* $\mathbf{L}$ *is finite and* T *is an* $\mathbf{L}$*-set of FAIs, then* $||A \Rightarrow B||_T$ *is the greatest degree* $a \in L$ *such that there is a* T'*-based* $\mathbf{L}^*$*-derivation DAG for* $A \Rightarrow a \otimes B$*, where* $T' = \{A \Rightarrow T(A \Rightarrow B) \otimes B \mid A, B \in L^Y \text{ and } T(A \Rightarrow B) \otimes B \not\subseteq A\}$*.*

Proof. Follows from Theorem 29 and (2.66). □

4.3 Computing closures

Considering the construction of T-based $\mathbf{L}^*$-derivation DAGs as an alternative proof technique not only can help visualize the inference from if-then rules but in addition, the construction of such DAGs yields algorithms for checking whether (and to what degree) $A \Rightarrow B$ semantically follows from a theory. Indeed, in order to check whether $||A \Rightarrow B||_T = 1$, we may proceed as follows:

Procedure 31 (Checking of full entailment)**.** *For any* T *and* $A \Rightarrow B$*:*

1. *Construct a* T*-based* $\mathbf{L}^*$*-derivation DAG* $\mathbf{D} = \langle V, \emptyset \rangle$ *with*

$$V = \{\langle y, A(y) \rangle \mid A(y) > 0\};$$

   If $V = \emptyset$*, put* $V = \{\langle y^\sharp, 0 \rangle\}$*, where* $y^\sharp$ *is the designated attribute, see Definition 23.*

2. *If* $\mathbf{D}(y) \geq B(y)$ *for all* $y \in R$*, stop and return "YES"; otherwise continue with step 3.*

3. *If* $\mathbf{D}$ *can be enlarged according to Definition 22 (case 2.), then enlarge* $\mathbf{D}$ *and continue with step 2.; otherwise return "NO".*

Provided that $\mathbf{L}$ and Y are finite, the three-step procedure always terminates and returns either "YES" or "NO". With respect to the semantic entailment from T, we can prove that the "YES" answer comes iff $A \Rightarrow B$ follows from T. Before we prove that we need some more observations and auxiliary notions. First, we look at combination of two (or more) T-based $\mathbf{L}^*$-derivation DAGs together to get a DAG with a combined yield which is greater than the yield of the input DAGs.

Lemma 32. *Let $\mathbf{D}_1 = \langle V_1, A_1\rangle$ and $\mathbf{D}_2 = \langle V_2, A_2\rangle$ be T-based $\mathbf{L}^*$-derivation DAGs with sets of initial vertices $I_1 \subseteq V_1$ and $I_2 \subseteq V_2$, respectively. Then there is a T-based $\mathbf{L}^*$-derivation DAG $\mathbf{D}_1 \cup \mathbf{D}_2 = \langle V, A\rangle$ with set of initial vertices*

$$\begin{aligned} I = &\{\langle y, a_1 \vee a_2\rangle \mid \langle y, a_1\rangle \in I_1 \text{ and } \langle y, a_2\rangle \in I_2\} \cup \\ &\{\langle y, a_1\rangle \mid \langle y, a_1\rangle \in I_1 \text{ and } \langle y, a_2\rangle \notin I_2 \text{ for all } a_2 \in L\} \cup \\ &\{\langle y, a_2\rangle \mid \langle y, a_2\rangle \in I_2 \text{ and } \langle y, a_1\rangle \notin I_1 \text{ for all } a_1 \in L\} \end{aligned} \tag{4.10}$$

such that $(\mathbf{D}_1 \cup \mathbf{D}_2)(y) \geq \mathbf{D}_1(y) \vee \mathbf{D}_2(y)$ for all $y \in R$.

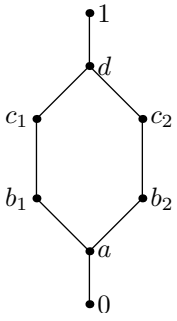
Proof. Initially, put $V = I$ as in (4.10) and $A = \emptyset$. Furthermore, let

$$\begin{aligned} W_1 &= \{\langle y, a\rangle \in V_1 \mid \langle y, b\rangle \notin I \text{ for all } b \geq a\}, \\ W_2 &= \{\langle y, a\rangle \in V_2 \mid \langle y, b\rangle \notin I \text{ for all } b \geq a\}. \end{aligned}$$

We are going to iteratively enlarge V and A by adding vertices (and arcs) based on vertices from W_1 and W_2. We start with vertices from W_1. Notice that from (4.10) it follows that $W_1 \cap I_1 = \emptyset$. During each step of the procedure, we ensure that for each vertex $\langle y, a\rangle \in V_1 \setminus W_1$, there is a vertex $\langle y, b\rangle \in V$ such that $a \leq b$. Initially, the condition follows directly from (4.10). Assume that $W_1 \neq \emptyset$. Since $W_1 \subseteq V_1$ and $\mathbf{D}_1$ is acyclic, W_1 satisfies the following

> *Property:* There is $\langle y, a\rangle \in W_1$ such that all arcs entering $\langle y, a\rangle$ in $\mathbf{D}_1$ leave from vertices $\langle y_i, a_i\rangle \in V_1 \setminus W_1$ $(i \in I)$.

Moreover, our assumption yields that for all those vertices $\langle y_i, a_i\rangle \in V_1 \setminus W_1$ $(i \in I)$ there are $\langle y_i, b_i\rangle \in V$ such that $a_i \leq b_i$ $(i \in I)$. Since $\langle y, a\rangle$ resulted from $\langle y_i, a_i\rangle \in V_1$ $(i \in I)$ considering some FAI from T, by considering the same FAI, we can compute the value of m and d, see (4.3) and (4.4), for vertices $\langle y_i, b_i\rangle \in V$ $(i \in I)$ and the attribute y. Note that due to the monotony of $\vee$, $\otimes$, *, and $\rightarrow$ in the second argument, we get $d \geq a$. If $d > m$, we may add $\langle y, d\rangle$ to V and add all arcs $\langle\langle y_i, b_i\rangle, \langle y, d\rangle\rangle$ to A $(i \in I)$. Otherwise, we left V and A unchanged. Then, we remove $\langle y, a\rangle$ from W_1. Since $d \geq a$, our initial assumption remains valid: for each $\langle y, a\rangle \in V_1 \setminus W_1$, there is $\langle y, b\rangle \in V$ such that $a \leq b$. Now, we may repeat the procedure until $W_1 = \emptyset$. Then, we continue the procedure with W_1, V_1, and I_1 replaced by W_2, V_2, and I_2 until $W_2 = \emptyset$. After that, we get $\mathbf{D}_1 \cup \mathbf{D}_2 = \langle V, A\rangle$ which is a T-based $\mathbf{L}^*$-derivation DAG that obviously satisfies $(\mathbf{D}_1 \cup \mathbf{D}_2)(y) \geq \mathbf{D}_1(y)$ for all $y \in R$ and $(\mathbf{D}_1 \cup \mathbf{D}_2)(y) \geq \mathbf{D}_2(y)$ for all $y \in R$. □



$\otimes$	0	a	b_1	b_2	c_1	c_2	d	1
0	0	0	0	0	0	0	0	0
a	0	a	a	a	a	a	a	a
b_1	0	a	b_1	a	b_1	a	b_1	b_1
b_2	0	a	a	b_2	a	b_2	b_2	b_2
c_1	0	a	b_1	a	b_1	a	b_1	c_1
c_2	0	a	a	b_2	a	b_2	b_2	c_2
d	0	a	b_1	b_2	b_1	b_2	d	d
1	0	a	b_1	b_2	c_1	c_2	d	1

$\rightarrow$	0	a	b_1	b_2	c_1	c_2	d	1
0	1	1	1	1	1	1	1	1
a	0	1	1	1	1	1	1	1
b_1	0	c_2	1	c_2	1	c_2	1	1
b_2	0	c_1	c_1	1	c_1	1	1	1
c_1	0	c_2	d	c_2	1	c_2	1	1
c_2	0	c_1	c_1	d	c_1	1	1	1
d	0	a	c_1	c_2	c_1	c_2	1	1
1	0	a	b_1	b_2	c_1	c_2	d	1

Figure 4.3: Nonlinear structure of truth degrees from Example 4

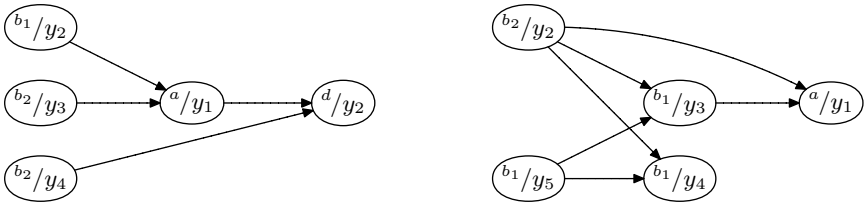


Figure 4.4: $\mathbf{L}^*$-derivation DAGs $\mathbf{D}_1$ and $\mathbf{D}_2$

Further in the thesis, we denote by $\mathbf{D}_1 \cup \mathbf{D}_2$ a DAG satisfying the conditions from Lemma 32 and call it a *union* of T-based $\mathbf{L}^*$-derivation DAGs $\mathbf{D}_1$ and $\mathbf{D}_2$. Note here that $\mathbf{D}_1 \cup \mathbf{D}_2$ satisfying conditions of Lemma 32 may not be given uniquely.

Remark 16. *One may be tempted to simplify the construction from Lemma 32 by taking the set-theoretic unions $V_1 \cup V_2$ and $A_1 \cup A_2$ for the sets of vertices and arcs of $\mathbf{D}_1 \cup \mathbf{D}_2$. However, the resulting structure may not be a T-based $\mathbf{L}^*$-derivation DAG (even if acyclic, it may not conform to the definition of a T-based $\mathbf{L}^*$-derivation DAG).*

Directly by observing (4.10), we get the following

Corollary 33. *If $\mathbf{D}_1$ and $\mathbf{D}_2$ are T-based $\mathbf{L}^*$-derivation DAGs which have the same set of initial vertices I, then the set of initial vertices of $\mathbf{D}_1 \cup \mathbf{D}_2$ is I.* □

In addition, in case of identical sets of initial vertices, $\mathbf{D}_1 \cup \mathbf{D}_2$ may be viewed as a DAG obtained from $\mathbf{D}_1$ by consecutively adding (some) vertices and arcs corresponding to vertices and arcs from $\mathbf{D}_2$, respectively, check the proof of Lemma 32. We further illustrate the construction of $\mathbf{D}_1 \cup \mathbf{D}_2$ in the following example.

Example 4. *For better illustration, we use a nonlinear structure of degrees as it is depicted by the Hasse diagram in Figure 4.3 (left); the adjoint operations $\otimes$ and $\rightarrow$ are defined by the tables. In addition, we assume that * is identity. Using this structure of degrees, we consider the following theory:*

$$T = \{\{{}^{b_2}/y_2, {}^{c_2}/y_5\} \Rightarrow \{{}^{c_1}/y_3, {}^{c_1}/y_4\}, \qquad (A1)$$

$$\{{}^{d}/y_3, {}^{d}/y_2\} \Rightarrow \{{}^{1}/y_1\}, \qquad (A2)$$

$$\{{}^{c_1}/y_1, {}^{c_2}/y_4\} \Rightarrow \{{}^{c_2}/y_2\}\}. \qquad (A3)$$

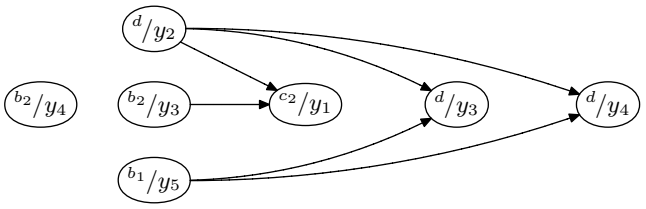


Figure 4.5: $\mathbf{L}^*$-derivation DAG $\mathbf{D}_1 \cup \mathbf{D}_2$

Let us consider T-based $\mathbf{L}^*$-derivation DAGs $\mathbf{D}_1$ and $\mathbf{D}_2$ for fuzzy attribute implications $\{^{b_1}/y_2, ^{b_2}/y_3, ^{b_2}/y_4\} \Rightarrow \{^{a}/y_1, ^{d}/y_2\}$ and $\{^{b_2}/y_2, ^{b_1}/y_5\} \Rightarrow \{^{b_1}/y_3, ^{b_1}/y_4, ^{a}/y_1\}$, respectively. The DAGs are depicted in Figure 4.4. $\mathbf{D}_1$ is constructed from the set of initial vertices $\{\langle y_2, b_1\rangle, \langle y_3, b_2\rangle, \langle y_4, b_2\rangle\}$ by adding vertices $\langle y_1, a\rangle$, $\langle y_2, d\rangle$ using FAIs (A2) and (A3), respectively. More precisely, when using (A2), we obtain (see (4.1)–(4.4)): $s_0 = \bigwedge \emptyset = 1$, $s_1 = \bigwedge\{d \to b_2, d \to b_1\} = c_2 \wedge c_1 = a$, $m = 0$, and $d = ((1 \wedge a)^* \otimes 1) \vee 0 = a$. When using (A3) we get: $s_0 = \bigwedge \emptyset = 1$, $s_1 = \bigwedge\{c_1 \to a, c_2 \to b_2\} = c_2$, $m = b_1$, and $d = ((1 \wedge c_2)^* \otimes c_2) \vee b_1 = b_2 \vee b_1$. $\mathbf{D}_2$ is constructed from $\{\langle y_2, b_2\rangle, \langle y_5, b_1\rangle\}$ by using (A1) twice for adding vertices $\langle y_3, b_1\rangle$, $\langle y_4, b_1\rangle$, and then using (A2) for adding $\langle y_1, a\rangle$.

The set of initial vertices for $\mathbf{D}_1 \cup \mathbf{D}_2$ given by Lemma 32 is $\{\langle y_2, d\rangle, \langle y_3, b_2\rangle, \langle y_4, b_2\rangle, \langle y_5, b_1\rangle\}$. Moreover, we have $W_1 = \{\langle y_1, a\rangle\}$ and $W_2 = \{\langle y_3, b_1\rangle, \langle y_4, b_1\rangle, \langle y_1, a\rangle\}$. During the process of construction of $\mathbf{D}_1 \cup \mathbf{D}_2$, we use (A2) once and (A1) twice for adding vertices $\langle y_1, c_2\rangle$, $\langle y_3, d\rangle$, and $\langle y_4, d\rangle$. The result is depicted in Figure 4.5.

Regarding the ability to enlarge a T-based $\mathbf{L}^*$-derivation DAG by adding vertices and arcs, we recognize so-called final DAGs which cannot be enlarged in this sense. The following definition introduces this notion formally.

Definition 34 (Final T-based $\mathbf{L}^*$-derivation DAG)**.** A T-based $\mathbf{L}^*$-derivation DAG is called final if there are no $E \Rightarrow F \in T$, attribute $y \in R$, and vertices $\langle y_1, a_1\rangle \in V, \ldots, \langle y_k, a_k\rangle \in V$ such that for s_0, s_1, m, d given by (4.1)–(4.4) we have $d > m$.

Final T-based $\mathbf{L}^*$-derivation DAGs possess the following property:

Lemma 35. *Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be T-based $\mathbf{L}^*$-derivation DAGs with the same set of initial vertices. If $\mathbf{D}_1$ is final, then $\mathbf{D}_1(y) \geq \mathbf{D}_2(y)$ for all $y \in R$.*

Proof. Let us consider there is $y \in R$ such that $\mathbf{D}_1(y) \not\geq \mathbf{D}_2(y)$. It suffices to show that $\mathbf{D}_1$ is not final. From Lemma 32 we get that $\mathbf{D}_1 \cup \mathbf{D}_2$ has the same set of initial vertices as $\mathbf{D}_1$ and $\mathbf{D}_2$. Furthermore, $\mathbf{D}_1(y) \not\geq \mathbf{D}_2(y)$ yields $(\mathbf{D}_1 \cup \mathbf{D}_2)(y) \geq \mathbf{D}_1(y) \vee \mathbf{D}_2(y) > \mathbf{D}_1(y)$. In addition, we may consider that $\mathbf{D}_1 \cup \mathbf{D}_2 \neq \mathbf{D}_1$ has been constructed by adding vertices and arcs to $\mathbf{D}_1$ (see the proof of Lemma 32), which means that $\mathbf{D}_1$ is not final. □

As a consequence, final T-based $\mathbf{L}^*$-derivation DAGs with the same sets of initial vertices are fully characterized by their yields:

Theorem 36. *Let $\mathbf{D}_1$ and $\mathbf{D}_2$ be T-based $\mathbf{L}^*$-derivation DAGs with the same set of initial vertices and let $\mathbf{D}_1$ be final. Then, $\mathbf{D}_2$ is final iff $\mathbf{D}_1(y) = \mathbf{D}_2(y)$ for all $y \in R$.*

Proof. From Lemma 35 it follows that if both $\mathbf{D}_1$ and $\mathbf{D}_2$ are final, then $\mathbf{D}_1(y) = \mathbf{D}_2(y)$ for all $y \in R$. Conversely, if $\mathbf{D}_1$ and $\mathbf{D}_2$ have the same yield for all $y \in R$ and $\mathbf{D}_1$ is final, from Lemma 35 it follows that one cannot extend $\mathbf{D}_2$ by another vertex because this would contradict that $\mathbf{D}_1$ is final, i.e., $\mathbf{D}_2$ must be final. □

We are now ready to prove correctness of Procedure 31.

Theorem 37. *Assuming* $\mathbf{L}$ *finite, for any* $A \Rightarrow B$ *and theory* T*, Procedure 31 terminates after finitely many steps and it returns "YES" iff* $||A \Rightarrow B||_T = 1$.

Proof. The termination is clear and follows from the finiteness of L^Y. If "YES" is returned, then $\mathbf{D}$ is a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$ and using Theorem 28, we get that $||A \Rightarrow B||_T = 1$. Conversely, let $||A \Rightarrow B||_T = 1$ and by contradiction, assume that "NO" has been returned. That means, for the last T-based $\mathbf{L}^*$-derivation DAG $\mathbf{D}$ there is $y \in R$ such that $B(y) \not\leq \mathbf{D}(y)$. Since $||A \Rightarrow B||_T = 1$, according to Theorem 28, there is a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$, denote it by $\mathbf{D}'$. Now, according to Lemma 32, the union $\mathbf{D} \cup \mathbf{D}'$ is a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$ since both $\mathbf{D}$ and $\mathbf{D}'$ have the same set of initial vertices. In addition, $(\mathbf{D} \cup \mathbf{D}')(y) > \mathbf{D}(y)$, which contradicts the fact that $\mathbf{D}$ was final, see Lemma 35. □

As a further consequence, which is easy to see, we get that if "NO" is answered by the above procedure, then the last $\mathbf{D}$ considered was final. The asymptotic worst-case time complexity of the procedure we have introduced for checking whether $A \Rightarrow B$ follows from T to a degree 1 is $O(p^2 n)$, where $p = |Y|$, and $n = |T|$ provided that the size of L is considered as a constant, cf. [6, 82].

The previous observations enable us to extend the procedure for computing syntactic closures for $\mathbf{L}$-sets of attributes. Let us recall that by a *closure of* A *under* T, denoted A_T^+, we mean the largest $\mathbf{L}$-set such that $T \vdash A \Rightarrow A_T^+$, see [13]. For every A and T, A_T^+ always exists, is uniquely given and has the following important property [13] provided that $\mathbf{L}$ is finite:

$$||A \Rightarrow B||_T = S\big(B, A_T^+\big). \tag{4.11}$$

The closure A_T^+ can be obtained from the yield of a final T-based $\mathbf{L}^*$-derivation DAG:

Theorem 38. *Let* $\mathbf{L}$ *be finite and* $\mathbf{D}$ *be a* T*-based* $\mathbf{L}^*$*-derivation DAG for* $A \Rightarrow B$*. Then* $\mathbf{D}$ *is final iff* $\mathbf{D}(y) = A_T^+(y)$ *for all* $y \in R$.

Proof. Since A_T^+ is known to be the greatest $\mathbf{L}$-set such that $T \vdash A \Rightarrow A_T^+$, there is a T-based $\mathbf{L}^*$-derivation DAG $\mathbf{D}'$ for $A \Rightarrow A_T^+$ which is final and $\mathbf{D}'(y) = A_T^+(y)$ for all $y \in R$ (otherwise we would get a contradiction to the fact that A_T^+ is the greatest $\mathbf{L}$-set such that $T \vdash A \Rightarrow A_T^+$). Now, we may apply Theorem 36. □

Owing to Theorem 38, in order to compute the degree $||A \Rightarrow B||_T$, it suffices to find a single final T-based $\mathbf{L}^*$-derivation DAG $\mathbf{D}$ for $A \Rightarrow B$, and apply (4.11) for A_T^+ determined from the yield of $\mathbf{D}$. We may formalize this computation by a modification of Procedure 31:

Procedure 39 (Compute entailment degree)**.** *For any T and $A \Rightarrow B$:*

1. *Step 1. remains the same as in Procedure 31.*
2. *If $\mathbf{D}$ can be enlarged according to Definition 22 (case 2.), then enlarge $\mathbf{D}$ and repeat step 2.; otherwise return $S(B, A_T^+)$, where $A_T^+(y) = \mathbf{D}(y)$ for all $y \in R$.*

Theorem 38 ensures that Procedure 39 is correct.

4.4 Illustrative example

For further illustration of the procedures discussed in this chapter, we present here an extended example in which we use the similarity-based database semantics of FAIs. Assume that a bank is keeping the following information about clients: city of residence (attribute c), age (attribute a), education (attribute e), job position (attribute j), salary (attribute s), loan amount (attribute l), account balance (attribute b), number of children (attribute ch), and insurance products (attribute i). Suppose that each attribute domain is equipped with a graded similarity relation. For instance, a similarity $\approx_c$ on the domain of cities may express the similarity of cities in terms of their size and location, $\approx_{ch}$ may express similarity of numbers of children in the account holder's household, $\approx_i$ can be based on type of insurance products the client has. In this setting, one may be interested in dependencies between values of the attributes. Since each domain is equipped with similarity, the dependencies may be satisfied (in data) to degrees and thus it is natural to express the dependencies by SBFDs.

Consider the set $L = [0, 1]$ together with Łukasiewicz operations and hedge $*$ being identity as a structure of truth degrees. Assume that the following set of SBFDs has been derived from a database (see [13] for a survey of methods):

$$
\begin{aligned}
T = \{&\{{}^{0.8}/_{c}, {}^{0.4}/_{a}, {}^{0.9}/_{j}\} \Rightarrow \{{}^{0.9}/_{s}\}, && \text{(A1)}\\
&\{{}^{0.9}/_{l}, {}^{0.8}/_{s}\} \Rightarrow \{{}^{0.8}/_{b}\}, && \text{(A2)}\\
&\{{}^{0.8}/_{ch}, {}^{0.8}/_{s}, {}^{0.8}/_{e}\} \Rightarrow \{{}^{0.8}/_{a}\}, && \text{(A3)}\\
&\{{}^{0.8}/_{e}, {}^{0.7}/_{a}\} \Rightarrow \{{}^{0.8}/_{ch}\} && \text{(A4)}\\
&\{{}^{0.7}/_{a}, {}^{0.7}/_{c}, {}^{0.8}/_{s}\} \Rightarrow \{{}^{0.8}/_{l}\}, && \text{(A5)}\\
&\{{}^{0.8}/_{b}, {}^{0.8}/_{a}, {}^{0.9}/_{l}, {}^{0.8}/_{ch}\} \Rightarrow \{{}^{0.8}/_{i}\}\}. && \text{(A6)}
\end{aligned}
$$

For example, (A5) can be read as follows: "If two clients have similar age at least to degree 0.7, live in cities which are similar at least to degree 0.7 and have salaries similar at least to degree 0.8, then their loan amount is similar at least to degree 0.8". Now consider $A = \{{}^{0.8}/_{c}, {}^{0.7}/_{j}, {}^{0.7}/_{e}, {}^{0.6}/_{ch}\}$ and $B = \{{}^{0.8}/_{i}, {}^{0.7}/_{l}\}$. Suppose we are interested in the degree to which $A \Rightarrow B$ follows from T (i.e., the degree $||A \Rightarrow B||_T$). According to Procedure 39, we construct a final T-based $\mathbf{L}^*$-derivation DAG $\mathbf{D}$ with initial vertices from A, determine the closure A_T^+, and compute the subsethood degree $S(B, A_T^+)$. The desired T-based $\mathbf{L}^*$-derivation DAG is depicted in Figure 4.6. In the process of its construction, we have used

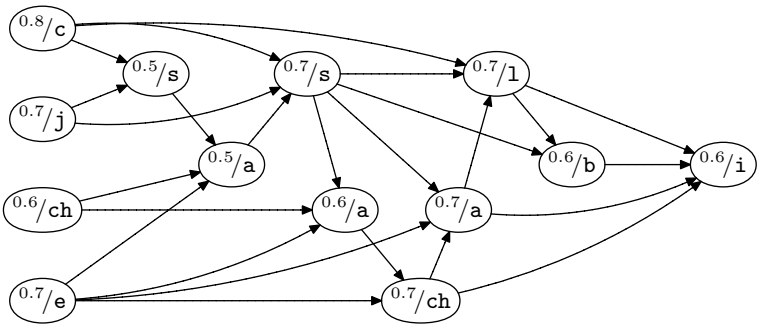


Figure 4.6: T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow A_T^+$.

(A1), (A3), (A1), (A3), (A4), (A3), (A5), (A2), and (A6) for adding vertices $\langle s, 0.5\rangle$, $\langle a, 0.5\rangle$, $\langle s, 0.7\rangle$, $\langle a, 0.6\rangle$, $\langle ch, 0.7\rangle$, $\langle a, 0.7\rangle$, $\langle l, 0.7\rangle$, $\langle b, 0.6\rangle$, and $\langle i, 0.6\rangle$, respectively. Notice that $\mathbf{D}$ is final and thanks to Theorem 38 we have

$$A_T^+ = \{^{0.8}/_{c}, {}^{0.7}/_{j}, {}^{0.7}/_{e}, {}^{0.7}/_{ch}, {}^{0.7}/_{s}, {}^{0.7}/_{a}, {}^{0.7}/_{l}, {}^{0.6}/_{b}, {}^{0.6}/_{i}\},$$

and thus

$$||A \Rightarrow B||_T = S(B, A_T^+) = (0.8 \to 0.6) \wedge (0.7 \to 0.7) = 0.8 \wedge 1 = 0.8.$$

The following sequence is a proof of $A \Rightarrow A_T^+$ constructed from the DAG in Figure 4.6. The proof uses derivation rules (Mul), (Ref), and (Acc) and has been constructed from the DAG according to the procedure from Theorem 27.

1) $\{^{0.8}/_{c}, {}^{0.7}/_{j}, {}^{0.7}/_{e}, {}^{0.6}/_{ch}\} \Rightarrow \{^{0.8}/_{c}, {}^{0.7}/_{j}, {}^{0.7}/_{e}, {}^{0.6}/_{ch}\}$
   (Ref)
2) $\{^{0.8}/_{c}, {}^{0.4}/_{a}, {}^{0.9}/_{j}\} \Rightarrow \{^{0.9}/_{s}\}$
   $\in T$
3) $\{^{0.4}/_{c}, {}^{0.5}/_{j}\} \Rightarrow \{^{0.5}/_{s}\}$
   (Mul) on 2)
4) $\{^{0.8}/_{c}, {}^{0.7}/_{j}, {}^{0.7}/_{e}, {}^{0.6}/_{ch}\} \Rightarrow \{^{0.8}/_{c}, {}^{0.7}/_{j}, {}^{0.7}/_{e}, {}^{0.6}/_{ch}, {}^{0.5}/_{s}\}$
   (Acc) on 1), 3)
5) $\{^{0.8}/_{ch}, {}^{0.8}/_{s}, {}^{0.8}/_{e}\} \Rightarrow \{^{0.8}/_{a}\}$
   $\in T$
6) $\{^{0.5}/_{ch}, {}^{0.5}/_{s}, {}^{0.5}/_{e}\} \Rightarrow \{^{0.5}/_{a}\}$
   (Mul) on 5)
7) $\{^{0.8}/_{c}, {}^{0.7}/_{j}, {}^{0.7}/_{e}, {}^{0.6}/_{ch}\} \Rightarrow \{^{0.8}/_{c}, {}^{0.7}/_{j}, {}^{0.7}/_{e}, {}^{0.6}/_{ch}, {}^{0.5}/_{s}, {}^{0.5}/_{a}\}$
   (Acc) on 4), 6)
8) $\{^{0.6}/_{c}, {}^{0.2}/_{a}, {}^{0.7}/_{j}\} \Rightarrow \{^{0.7}/_{s}\}$
   (Mul) on 2)
9) $\{^{0.8}/_{c}, {}^{0.7}/_{j}, {}^{0.7}/_{e}, {}^{0.6}/_{ch}\} \Rightarrow \{^{0.8}/_{c}, {}^{0.7}/_{j}, {}^{0.7}/_{e}, {}^{0.6}/_{ch}, {}^{0.7}/_{s}, {}^{0.5}/_{a}\}$
   (Acc) on 7), 8)

10) $\{^{0.6}/_{ch}, ^{0.6}/_{s}, ^{0.6}/_{e}\} \Rightarrow \{^{0.6}/_{a}\}$
(Mul) on 5)
11) $\{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.6}/_{ch}\} \Rightarrow \{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.6}/_{ch}, ^{0.7}/_{s}, ^{0.6}/_{a}\}$
(Acc) on 9), 10)
12) $\{^{0.8}/_{e}, ^{0.7}/_{a}\} \Rightarrow \{^{0.8}/_{ch}\}$
$\in T$
13) $\{^{0.7}/_{e}, ^{0.6}/_{a}\} \Rightarrow \{^{0.7}/_{ch}\}$
(Mul) on 12)
14) $\{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.6}/_{ch}\} \Rightarrow \{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.7}/_{ch}, ^{0.7}/_{s}, ^{0.6}/_{a}\}$
(Acc) on 11), 13)
15) $\{^{0.7}/_{ch}, ^{0.7}/_{s}, ^{0.7}/_{e}\} \Rightarrow \{^{0.7}/_{a}\}$
(Mul) on 5)
16) $\{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.6}/_{ch}\} \Rightarrow \{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.7}/_{ch}, ^{0.7}/_{s}, ^{0.7}/_{a}\}$
(Acc) on 14), 15)
17) $\{^{0.7}/_{a}, ^{0.7}/_{c}, ^{0.8}/_{s}\} \Rightarrow \{^{0.8}/_{l}\}$
$\in T$
18) $\{^{0.6}/_{a}, ^{0.6}/_{c}, ^{0.7}/_{s}\} \Rightarrow \{^{0.7}/_{l}\}$
(Mul) on 17)
19) $\{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.6}/_{ch}\} \Rightarrow \{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.7}/_{ch}, ^{0.7}/_{s}, ^{0.7}/_{a}, ^{0.7}/_{l}\}$
(Acc) on 16), 18)
20) $\{^{0.9}/_{l}, ^{0.8}/_{s}\} \Rightarrow \{^{0.8}/_{b}\}$
$\in T$
21) $\{^{0.7}/_{l}, ^{0.6}/_{s}\} \Rightarrow \{^{0.6}/_{b}\}$
(Mul) on 20)
22) $\{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.6}/_{ch}\} \Rightarrow$
$\Rightarrow \{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.7}/_{ch}, ^{0.7}/_{s}, ^{0.7}/_{a}, ^{0.7}/_{l}, ^{0.6}/_{b}\}$
(Acc) on 19), 21)
23) $\{^{0.8}/_{b}, ^{0.8}/_{a}, ^{0.9}/_{l}, ^{0.8}/_{ch}\} \Rightarrow \{^{0.8}/_{i}\}$
$\in T$
24) $\{^{0.6}/_{b}, ^{0.6}/_{a}, ^{0.7}/_{l}, ^{0.6}/_{ch}\} \Rightarrow \{^{0.6}/_{i}\}$
(Mul) on 23)
25) $\{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.6}/_{ch}\} \Rightarrow$
$\Rightarrow \{^{0.8}/_{c}, ^{0.7}/_{j}, ^{0.7}/_{e}, ^{0.7}/_{ch}, ^{0.7}/_{s}, ^{0.7}/_{a}, ^{0.7}/_{l}, ^{0.6}/_{b}, ^{0.6}/_{i}\}$
(Acc) on 22), 24)

Notice that if we reorder the formulas in the preceding proof (see the proof of Theorem 27), we obtain an MRAP sequence for $A \Rightarrow A_T^+$ which corresponds to the DAG. Notice that (Pro) is not needed in this case.

4.5 Conclusions

We have presented a graph-based inference methods for graded if-then rules, particularly for SBFDs from Section 2.3.3. We have introduced a notion of a T-based $\mathbf{L}^*$-derivation directed acyclic graph (DAG) which generalizes the ordinary notion of a T-based derivation DAG from [81]. The main results show that degrees of semantic entailment of if-then rules from collections of other if-then rules can be characterized by the existence of such directed acyclic graphs. We have proved the result by showing correspondences between the constructions of T-based $\mathbf{L}^*$-derivation DAGs and proofs from T.

Chapter 5

Sensitivity analysis for similarity-based functional dependencies

In this chapter we look at similarity estimates for SBFDs (given by Equation (2.62)) in ranked data tables (RDTs) over domains with similarities (see Section 2.3). We answer some natural questions such as: What is the relationship between $||A \Rightarrow B||_{\mathcal{D}_1}$ and $||A \Rightarrow B||_{\mathcal{D}_2}$ in terms of similarity of RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$? Or what can we say about the truth degrees $||A \Rightarrow B_1||_{\mathcal{D}}$ and $||A \Rightarrow B_2||_{\mathcal{D}}$ in terms of similarity of B_1 and B_2. The first problem we discuss in this chapter is how to assess similarity of two ranked data tables.

5.1 Rank-based similarity

In this section, we introduce a notion of a similarity and a related notion of a graded containment (subsethood) of RDTs on the same relation scheme R. As in the case of domain similarities, the similarity of RDTs is expressed by degrees from the complete residuated lattice $\mathbf{L}$.

The rank-based similarity of RDTs which is based on the idea that RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$ (on the same relation scheme R) are similar iff for each tuple $r \in \mathrm{Tupl}(R)$, ranks $\mathcal{D}_1(r)$ and $\mathcal{D}_2(r)$ are similar degrees from $\mathbf{L}$. Similarity of degrees from $\mathbf{L}$ can be expressed by a biresiduum (2.2). Since we are interested in assessing similarity of $\mathcal{D}_1(r)$ and $\mathcal{D}_2(r)$ for all possible tuples r, we may define the *similarity* $E(\mathcal{D}_1, \mathcal{D}_2)$ of RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$ as an infimum which goes over all tuples:

$$E(\mathcal{D}_1, \mathcal{D}_2) = \bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \leftrightarrow \mathcal{D}_2(r)\big). \tag{5.1}$$

If $\mathcal{D}_1$ and $\mathcal{D}_2$ are results of queries Q_1 and Q_2, then (5.1) can be interpreted as a degree to which the following proposition is true: "Each tuple matches query Q_1 if and only if it matches query Q_2." Thus, $\mathcal{D}_1$ and $\mathcal{D}_2$ are considered similar if they represent similar answers to queries.

Remark 17. *Note that from the computational point of view,* (5.1) *involves a general infimum which goes over all tuples in* $\mathrm{Tupl}(R)$*. If* R *contains an attribute* $y \in R$ *whose domain* D_y *is infinite, so is* $\mathrm{Tupl}(R)$*. Thus, in order to evaluate* (5.1)*, one has to go over infinitely many tuples. Nevertheless, determining* $E(\mathcal{D}_1, \mathcal{D}_2)$ *is tractable since only finitely many tuples have nonzero ranks in* $\mathcal{D}_1$ *and* $\mathcal{D}_2$*. Indeed, from properties of* $\leftrightarrow$*, we can easily see that in* (5.1)*, only for finitely many tuples* $r \in \mathrm{Tupl}(R)$*, the value of* $\mathcal{D}_1(r) \leftrightarrow \mathcal{D}_2(r)$ *is nonzero and for all* $r \in \mathrm{Tupl}(R)$ *such that* $\mathcal{D}_1(r) = \mathcal{D}_2(r) = 0$*, we have* $\mathcal{D}_1(r) \leftrightarrow \mathcal{D}_2(r) = 1$ *which does not contribute to the degree* $E(\mathcal{D}_1, \mathcal{D}_2)$*. Hence, the right-hand side of* (5.1) *can be rewritten as*

$$\bigwedge\{\mathcal{D}_1(r) \leftrightarrow \mathcal{D}_2(r) \,|\, \mathcal{D}_1(r) > 0 \text{ or } \mathcal{D}_2(r) > 0\}.$$

Also note that for $a_\emptyset$ *and* $b_\emptyset$*,* $E(a_\emptyset, b_\emptyset) = a \leftrightarrow b$*.*

An alternative (but equivalent) way to define similarity of RDTs is the following: we first formalize a degree $S(\mathcal{D}_1, \mathcal{D}_2)$ to which $\mathcal{D}_1$ is included in $\mathcal{D}_2$. The motivation for having degrees of inclusion comes from considering a rank-aware generalization of the subsethood of classic relations. In case of RDTs, we can say that $\mathcal{D}_1$ is fully included in $\mathcal{D}_2$ iff, for each tuple r, the rank $\mathcal{D}_2(r)$ is at least as high as the rank $\mathcal{D}_1(r)$. Notice that in the ordinary case, this is exactly how one defines the ordinary subsethood relation "$\subseteq$". Considering general degrees of inclusion (subsethood), a degree $S(\mathcal{D}_1, \mathcal{D}_2)$ to which $\mathcal{D}_1$ is *included* in $\mathcal{D}_2$ can be defined as follows:

$$S(\mathcal{D}_1, \mathcal{D}_2) = \textstyle\bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \to \mathcal{D}_2(r)\big). \tag{5.2}$$

Described verbally, $S(\mathcal{D}_1, \mathcal{D}_2)$ is a degree to which "each tuple matches Q_2 at least to the degree to which it matches Q_1" provided that $\mathcal{D}_1$ and $\mathcal{D}_2$ are interpreted as results of queries Q_1 and Q_2, respectively. Similar observations as in Remark 17 apply: $S(\mathcal{D}_1, \mathcal{D}_2)$ can be computed considering only finitely many tuples from $\mathrm{Tupl}(R)$. In this case, it suffices to go over tuples which have nonzero ranks in $\mathcal{D}_1$ because $0 \to a = 1$ for all $a \in L$.

Remark 18. *By a slight abuse of notation, we denote the fact* $S(\mathcal{D}_1, \mathcal{D}_2) = 1$ *by* $\mathcal{D}_1 \subseteq \mathcal{D}_2$*. Observe that* $S(\mathcal{D}_1, \mathcal{D}_2) = 1$ *iff* $\mathcal{D}_1(r) \leq \mathcal{D}_2(r)$ *for all* $r \in \mathrm{Tupl}(R)$*. Analogously, we denote* $E(\mathcal{D}_1, \mathcal{D}_2) = 1$ *by* $\mathcal{D}_1 = \mathcal{D}_2$*. In this case,* $E(\mathcal{D}_1, \mathcal{D}_2) = 1$ *iff* $\mathcal{D}_1(r) = \mathcal{D}_2(r)$ *for all* $r \in \mathrm{Tupl}(R)$*.*

It is easy to prove [9] that (5.1) and (5.2) satisfy:

$$E(\mathcal{D}_1, \mathcal{D}_2) = S(\mathcal{D}_1, \mathcal{D}_2) \wedge S(\mathcal{D}_2, \mathcal{D}_1). \tag{5.3}$$

Note that E and S defined by (5.1) and (5.2) are known as degrees of similarity (2.34) and subsethood (2.33) (in this case, the fuzzy relations are RDTs).

Remark 19. *(a) An interesting point is the relationship between the notions of domain similarity and similarity defined by* (5.1)*. In fact,* (5.1) *can be seen as a definition of*

a similarity on a domain of all RDTs on R. This follows from the fact that E defined by (5.1) *is obviously reflexive and symmetric (in addition, it is separating and* $\otimes$*-transitive). As a consequence, if one requires RDTs as values of attributes (e.g., as in [54]),* (5.1) *can be used to measure similarity on domains of RDTs. We consider this to be an important aspect of the model, showing its universality.*

(b) Note that if the complete residuated lattice $\mathbf{L}$ *becomes the two-element Boolean algebra, the notions of degree of similarity and inclusion become the ordinary (bivalent) notions of equality and inclusion of data tables, meaning that* $S(\mathcal{D}_1, \mathcal{D}_2) = 1$ *iff each tuple from* $\mathcal{D}_1$ *is also in* $\mathcal{D}_2$*, and* $S(\mathcal{D}_1, \mathcal{D}_2) = 0$ *otherwise (*$\mathcal{D}_1$ *contains a tuple which is not in* $\mathcal{D}_2$*). Analogously for E.*

5.2 Similarity estimates for similarity-based FD

We have seen how to define the similarity of two RDTs on the same relation scheme. An interesting question regarding the validity of SBFDs is: What can we say about the truth degree of $A \Rightarrow B$ in similar RDTs? Before we answer this question, we present a Lemma that establishes the relationship between the similarity of two tuples in two different RDTs on the same relation scheme. Recall from (2.3) that $a^2 = a \otimes a$.

Lemma 40. *For RDTs* $\mathcal{D}_1$*,* $\mathcal{D}_2$ *on the same relation scheme R and for any tuples* $r_1, r_2 \in \mathrm{Tupl}(R)$*:*

$$S(\mathcal{D}_1, \mathcal{D}_2)^2 \otimes (r_1(A) \approx_{\mathcal{D}_2} r_2(A)) \leq r_1(A) \approx_{\mathcal{D}_1} r_2(A), \tag{5.4}$$

$$(S(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes (r_1(A) \approx_{\mathcal{D}_2} r_2(A))^* \leq (r_1(A) \approx_{\mathcal{D}_1} r_2(A))^*. \tag{5.5}$$

Proof. Using the fact that for any $a, b, c \in L$

$$(a \otimes b) \to c = a \to (b \to c) = b \to (a \to c)$$

together with (2.16) and (5.2) we have

$$S(\mathcal{D}_1, \mathcal{D}_2)^2 \otimes (r_1(A) \approx_{\mathcal{D}_2} r_2(A)) =$$

$$S(\mathcal{D}_1, \mathcal{D}_2)^2 \otimes \Big((\mathcal{D}_2(r_1) \otimes \mathcal{D}_2(r_2)) \to \bigwedge_{y \in R} (A(y) \to r_1(y) \approx_y r_2(y))\Big) =$$

$$S(\mathcal{D}_1, \mathcal{D}_2)^2 \otimes \Big(\mathcal{D}_2(r_1) \to \big(\mathcal{D}_2(r_2) \to \bigwedge_{y \in R} (A(y) \to r_1(y) \approx_y r_2(y))\big)\Big) \leq$$

$$S(\mathcal{D}_1, \mathcal{D}_2) \otimes (\mathcal{D}_1(r_1) \to \mathcal{D}_2(r_1)) \otimes \Big(\mathcal{D}_2(r_1) \to \big(\mathcal{D}_2(r_2) \to \bigwedge_{y \in R} (A(y) \to r_1(y) \approx_y r_2(y))\big)\Big) \leq$$

$$S(\mathcal{D}_1, \mathcal{D}_2) \otimes \Big(\mathcal{D}_1(r_1) \to \big(\mathcal{D}_2(r_2) \to \bigwedge_{y \in R} (A(y) \to r_1(y) \approx_y r_2(y))\big)\Big) =$$

$$S(\mathcal{D}_1, \mathcal{D}_2) \otimes \Big(\mathcal{D}_2(r_2) \to \big(\mathcal{D}_1(r_1) \to \bigwedge_{y \in R} (A(y) \to r_1(y) \approx_y r_2(y))\big)\Big) \leq$$

$$(\mathcal{D}_1(r_2) \to \mathcal{D}_2(r_2)) \otimes \Big(\mathcal{D}_2(r_2) \to \big(\mathcal{D}_1(r_1) \to \bigwedge_{y \in R} (A(y) \to r_1(y) \approx_y r_2(y))\big)\Big) \leq$$

$$\mathcal{D}_1(r_2) \to \big(\mathcal{D}_1(r_1) \to \bigwedge_{y\in R} (A(y) \to r_1(y) \approx_y r_2(y))\big) =$$
$$(\mathcal{D}_1(r_2) \otimes \mathcal{D}_1(r_1)) \to \bigwedge_{y\in R} (A(y) \to r_1(y) \approx_y r_2(y)) = r_1(A) \approx_{\mathcal{D}_1} r_2(A).$$

The second inequality follows from the first one using $a^* \otimes b^* \leq (a \otimes b)^*$ twice. □

The following Theorem shows the relationship between the similarity of two RDTs $\mathcal{D}_1$, $\mathcal{D}_2$ and the degrees to which they satisfy a SBFD $A \Rightarrow B$.

Theorem 41. *For RDTs $\mathcal{D}_1$, $\mathcal{D}_2$ on the same relation scheme R we have*

$$(S(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes S(\mathcal{D}_2, \mathcal{D}_1)^2 \leq ||A \Rightarrow B||_{\mathcal{D}_1} \to ||A \Rightarrow B||_{\mathcal{D}_2}, \tag{5.6}$$
$$(E(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes E(\mathcal{D}_2, \mathcal{D}_1)^2 \leq ||A \Rightarrow B||_{\mathcal{D}_1} \leftrightarrow ||A \Rightarrow B||_{\mathcal{D}_2}. \tag{5.7}$$

Proof. Using adjointness the Equation (5.6) is equivalent to

$$(S(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes S(\mathcal{D}_2, \mathcal{D}_1)^2 \otimes ||A \Rightarrow B||_{\mathcal{D}_1} \leq ||A \Rightarrow B||_{\mathcal{D}_2}.$$

Thus it suffices to show that

$$(S(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes S(\mathcal{D}_2, \mathcal{D}_1)^2 \otimes ||A \Rightarrow B||_{\mathcal{D}_1} \leq (r_1(A) \approx_{\mathcal{D}_2} r_2(A))^* \to (r_1(B) \approx_{\mathcal{D}_2} r_2(B))$$

is true for any $r_1, r_2 \in \mathrm{Tupl}(R)$. Which is further equivalent to

$$(S(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes S(\mathcal{D}_2, \mathcal{D}_1)^2 \otimes ||A \Rightarrow B||_{\mathcal{D}_1} \otimes (r_1(A) \approx_{\mathcal{D}_2} r_2(A))^* \leq (r_1(B) \approx_{\mathcal{D}_2} r_2(B)).$$

The last inequality is indeed true, for any $r_1, r_2 \in \mathrm{Tupl}(R)$ we have

$$\begin{aligned}
&(S(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes S(\mathcal{D}_2, \mathcal{D}_1)^2 \otimes ||A \Rightarrow B||_{\mathcal{D}_1} \otimes (r_1(A) \approx_{\mathcal{D}_2} r_2(A))^* = \\
&(S(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes (r_1(A) \approx_{\mathcal{D}_2} r_2(A))^* \otimes S(\mathcal{D}_2, \mathcal{D}_1)^2 \otimes ||A \Rightarrow B||_{\mathcal{D}_1} \leq \\
&(r_1(A) \approx_{\mathcal{D}_1} r_2(A))^* \otimes S(\mathcal{D}_2, \mathcal{D}_1)^2 \otimes ||A \Rightarrow B||_{\mathcal{D}_1} \leq \\
&S(\mathcal{D}_2, \mathcal{D}_1)^2 \otimes (r_1(A) \approx_{\mathcal{D}_1} r_2(A))^* \otimes \big((r_1(A) \approx_{\mathcal{D}_1} r_2(A))^* \to (r_1(B) \approx_{\mathcal{D}_1} r_2(B))\big) \leq \\
&S(\mathcal{D}_2, \mathcal{D}_1)^2 \otimes (r_1(B) \approx_{\mathcal{D}_1} r_2(B)) \leq (r_1(B) \approx_{\mathcal{D}_2} r_2(B)),
\end{aligned}$$

using Lemma 40 and (2.13). The Equation (5.7) is a consequence of (5.6), monotony of hedge, and of the fact that for any degrees $a_{1i} \in L$, $a_{2i} \in L$ where $i = 1, \ldots, n$, we have

$$\bigotimes_{i=1}^n (a_{1i} \wedge a_{2i}) \leq \bigotimes_{i=1}^n a_{1i} \wedge \bigotimes_{i=1}^n a_{2i}. \tag{5.8}$$

Indeed:

$$\begin{aligned}
&(E(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes E(\mathcal{D}_2, \mathcal{D}_1)^2 = \big((S(\mathcal{D}_1, \mathcal{D}_2) \wedge S(\mathcal{D}_2, \mathcal{D}_1))^*\big)^2 \otimes E(\mathcal{D}_2, \mathcal{D}_1)^2 \leq \\
&\big(S(\mathcal{D}_1, \mathcal{D}_2)^* \wedge S(\mathcal{D}_2, \mathcal{D}_1)^*\big)^2 \otimes \big(S(\mathcal{D}_2, \mathcal{D}_1) \wedge S(\mathcal{D}_1, \mathcal{D}_2)\big)^2 \leq \\
&\big((S(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \wedge (S(\mathcal{D}_2, \mathcal{D}_1)^*)^2\big) \otimes \big((S(\mathcal{D}_2, \mathcal{D}_1))^2 \wedge (S(\mathcal{D}_1, \mathcal{D}_2))^2\big) \leq \\
&\big((S(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes (S(\mathcal{D}_2, \mathcal{D}_1))^2\big) \wedge \big((S(\mathcal{D}_2, \mathcal{D}_1)^*)^2 \otimes (S(\mathcal{D}_1, \mathcal{D}_2))^2 \leq \\
&(||A \Rightarrow B||_{\mathcal{D}_1} \to ||A \Rightarrow B||_{\mathcal{D}_2}) \wedge (||A \Rightarrow B||_{\mathcal{D}_2} \to ||A \Rightarrow B||_{\mathcal{D}_1}) = \\
&||A \Rightarrow B||_{\mathcal{D}_1} \leftrightarrow ||A \Rightarrow B||_{\mathcal{D}_2}.
\end{aligned}$$

□

If we take an identity for a hedge, the left hand side of the Equation (5.7) can be simplified.

Corollary 42. *For RDTs $\mathcal{D}_1$, $\mathcal{D}_2$ on the same relation scheme R and for hedge being identity we have*

$$E(\mathcal{D}_1, \mathcal{D}_2)^4 \leq ||A \Rightarrow B||_{\mathcal{D}_1} \leftrightarrow ||A \Rightarrow B||_{\mathcal{D}_2}.$$

We now turn our attention to hedges. Since a hedge is used as a parameter in the definition of SBFD, the natural question is how the truth degree of SBFD $A \Rightarrow B$ when hedge $*_1$ is used differs from the truth degree of the same SBFD when hedge $*_2$ is used. In order to emphasize the hedge used in the definition of SBFD, we will employ the following notation: $||A \Rightarrow B||^*_{\mathcal{D}}$. First of all, we need to capture the similarity of two hedges:

Definition 43 ([16])**.** *For hedges $*_1, *_2$ on $\mathbf{L}$ put*

$$(*_1 \preceq *_2) = \bigwedge_{a \in L} (a^{*_1} \rightarrow a^{*_2}), \tag{5.9}$$

$$(*_1 \approx *_2) = \bigwedge_{a \in L} (a^{*_1} \leftrightarrow a^{*_2}). \tag{5.10}$$

The Equation (5.10) can be interpreted as a degree to which hedges $*_1$ and $*_2$ yield similar results. More precisely, (5.10) is a true degree of of the following formula: "for each $a \in L$: the result of a^{*_1} is similar to the result of a^{*_2}." Analogously, (5.9) can be interpreted as a degree to which $*_1$ is stronger than $*_2$.

Theorem (44) shows that "if $A \Rightarrow B$ is true using hedge $*_2$ and if hedge $*_1$ is stronger than $*_2$, then $A \Rightarrow B$ is true using hedge $*_1$" and that "if the hedges $*_1$ and $*_2$ are similar, then the degrees to which $A \Rightarrow B$ is true using hedge $*_1$ and hedge $*_2$ are similar".

Theorem 44. *Let $A, B \in \mathbf{L}^R$ and let $*_1$, $*_2$ be two hedges on $\mathbf{L}$. Then for any RDT $\mathcal{D}$ on R we have:*

$$(*_1 \preceq *_2) \leq ||A \Rightarrow B||^{*_2}_{\mathcal{D}} \rightarrow ||A \Rightarrow B||^{*_1}_{\mathcal{D}}, \tag{5.11}$$

$$(*_1 \approx *_2) \leq ||A \Rightarrow B||^{*_2}_{\mathcal{D}} \leftrightarrow ||A \Rightarrow B||^{*_1}_{\mathcal{D}}. \tag{5.12}$$

Proof. The first inequality is true iff

$$(*_1 \preceq *_2) \otimes ||A \Rightarrow B||^{*_2}_{\mathcal{D}} \leq ||A \Rightarrow B||^{*_1}_{\mathcal{D}}$$

holds. Using (2.21) we have:

$$\begin{aligned}
&(*_1 \preceq *_2) \otimes ||A \Rightarrow B||^{*_2}_{\mathcal{D}} = \\
&(*_1 \preceq *_2) \otimes \bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} (r_1(A) \approx_D r_2(A))^{*_2} \rightarrow (r_1(B) \approx_D r_2(B)) \leq \\
&\bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} (*_1 \preceq *_2) \otimes \big((r_1(A) \approx_D r_2(A))^{*_2} \rightarrow (r_1(B) \approx_D r_2(B))\big) =
\end{aligned}$$

$$\bigwedge_{r_1,r_2\in \mathrm{Tupl}(R)} (\bigwedge_{a\in L}(a^{*_1}\to a^{*_2}))\otimes \big((r_1(A)\approx_D r_2(A))^{*_2}\to r_1(B)\approx_D r_2(B)\big).$$

Now observe that

$$\bigwedge_{a\in L}(a^{*_1}\to a^{*_2})\leq (r_1(A)\approx_D r_2(A))^{*_1}\to (r_1(A)\approx_D r_2(A))^{*_2}$$

which together with $\otimes$-transitivity of residuum (2.16) yields:

$$\begin{aligned}
&(*_1\preceq *_2)\otimes ||A\Rightarrow B||^{*_2}_{\mathcal{D}}\leq\\
&\bigwedge_{r_1,r_2\in \mathrm{Tupl}(R)} \Big(\big((r_1(A)\approx_D r_2(A))^{*_1}\to (r_1(A)\approx_D r_2(A))^{*_2}\big)\otimes\\
&\quad\otimes\big((r_1(A)\approx_D r_2(A))^{*_2}\to r_1(B)\approx_D r_2(B)\big)\Big)\leq\\
&\bigwedge_{r_1,r_2\in \mathrm{Tupl}(R)} (r_1(A)\approx_D r_2(A))^{*_1}\to (r_1(B)\approx_D r_2(B))=||A\Rightarrow B||^{*_1}_{\mathcal{D}}
\end{aligned}$$

finishing the proof of (5.11). The inequality (5.12) is a consequence of (5.11). □

The previous theorem showed how the truth degree of SBFD $A\Rightarrow B$ changes if we change the hedge. The next problem we want to tackle is how the truth degree of $A\Rightarrow B$ depends on the truth degrees prescribed by the $\mathbf{L}$-sets A, B. First, we present an auxiliary lemma, which we will use in the subsequent proofs. The Lemma 45 states that if two tuples are similar on the attributes A and if B is subset of A, then they are similar on B as well.

Lemma 45. *Let $A, B\in \mathbf{L}^R$ be fuzzy sets of attributes. For any RDT $\mathcal{D}$ on R and any tuples $r_1, r_2\in\mathcal{D}$ we have:*

$$S(B,A)\otimes (r_1(A)\approx_D r_2(A))\leq (r_1(B)\approx_D r_2(B)). \tag{5.13}$$

Proof. Using (2.15) we get

$$\begin{aligned}
&S(B,A)\otimes (r_1(A)\approx_D r_2(A))=\\
&S(B,A)\otimes \big((\mathcal{D}(r_1)\otimes\mathcal{D}(r_2))\to \bigwedge_{y\in R}(A(y)\to r_1(y)\approx_y r_2(y))\big)\leq\\
&(\mathcal{D}(r_1)\otimes\mathcal{D}(r_2))\to \big(S(B,A)\otimes \bigwedge_{y\in R}(A(y)\to r_1(y)\approx_y r_2(y))\big).
\end{aligned}$$

Now observe that using (2.21), (2.15) and (2.16) we have

$$\begin{aligned}
&S(B,A)\otimes \bigwedge_{y\in R}(A(y)\to r_1(y)\approx_y r_2(y))\leq\\
&\bigwedge_{y\in R}\big(S(B,A)\otimes (A(y)\to r_1(y)\approx_y r_2(y))\big)\leq\\
&\bigwedge_{y\in R}\big((B(y)\to A(y))\otimes (A(y)\to r_1(y)\approx_y r_2(y))\big)\leq
\end{aligned}$$

$$\bigwedge_{y\in R}(B(y)\to r_1(y)\approx_y r_2(y)),$$

which together with isotony of $\to$ in the second argument yields

$$(\mathcal{D}(r_1)\otimes\mathcal{D}(r_2))\to\big(S(B,A)\otimes\bigwedge_{y\in R}(A(y)\to r_1(y)\approx_y r_2(y))\big)\leq$$
$$(\mathcal{D}(r_1)\otimes\mathcal{D}(r_2))\to\big(\bigwedge_{y\in R}(B(y)\to r_1(y)\approx_y r_2(y))\big).$$

Putting the previous observations together we finally obtain

$$\begin{aligned}&S(B,A)\otimes(r_1(A)\approx_D r_2(A))\leq\\&(\mathcal{D}(r_1)\otimes\mathcal{D}(r_2))\to\big(\bigwedge_{y\in R}(B(y)\to r_1(y)\approx_y r_2(y))\big)=\\&(r_1(B)\approx_D r_2(B)).\end{aligned}$$

□

We are now ready to show how the validity of $A\Rightarrow B$ will change, if we replace the sets of attributes A and B by similar ones.

Lemma 46. *Let $A,B_1,B_2\in\mathbf{L}^R$ be fuzzy sets of attributes. For any RDT $\mathcal{D}$ on R we have*

$$S(B_2,B_1)\otimes||A\Rightarrow B_1||_{\mathcal{D}}\leq||A\Rightarrow B_2||_{\mathcal{D}}. \tag{5.14}$$

Proof. Using the Lemma 45 and (2.15) we observe that

$$\begin{aligned}&S(B_2,B_1)\otimes||A\Rightarrow B_1||_{\mathcal{D}}=\\&S(B_2,B_1)\otimes\bigwedge_{r_1,r_2\in\mathrm{Tupl}(R)}\big((r_1(A)\approx_{\mathcal{D}} r_2(A))^*\to(r_1(B_1)\approx_{\mathcal{D}} r_2(B_1))\big)\leq\\&\bigwedge_{r_1,r_2\in\mathrm{Tupl}(R)}S(B_2,B_1)\otimes\big((r_1(A)\approx_{\mathcal{D}} r_2(A))^*\to(r_1(B_1)\approx_{\mathcal{D}} r_2(B_1))\big)\leq\\&\bigwedge_{r_1,r_2\in\mathrm{Tupl}(R)}(r_1(A)\approx_{\mathcal{D}} r_2(A))^*\to\big(S(B_2,B_1)\otimes(r_1(B_1)\approx_{\mathcal{D}} r_2(B_1))\big)\leq\\&\bigwedge_{r_1,r_2\in\mathrm{Tupl}(R)}(r_1(A)\approx_{\mathcal{D}} r_2(A))^*\to(r_1(B_2)\approx_{\mathcal{D}} r_2(B_2))\big)=||A\Rightarrow B_2||_{\mathcal{D}}.\end{aligned}$$

□

Lemma 47. *Let $A_1,A_2,B\in\mathbf{L}^R$ be fuzzy sets of attributes. For any RDT $\mathcal{D}$ on R we have*

$$S(A_1,A_2)^*\otimes||A_1\Rightarrow B||_{\mathcal{D}}\leq||A_2\Rightarrow B||_{\mathcal{D}}. \tag{5.15}$$

Proof. Using adjointness, (5.15) is equivalent to

$$||A_1 \Rightarrow B||_{\mathcal{D}} \leq S(A_1, A_2)^* \rightarrow ||A_2 \Rightarrow B||_{\mathcal{D}}.$$

From (2.18) and (2.14) we have:

$$\begin{aligned}
&S(A_1, A_2)^* \rightarrow ||A_2 \Rightarrow B||_{\mathcal{D}} = \\
&S(A_1, A_2)^* \rightarrow \bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} \big((r_1(A_2) \approx_{\mathcal{D}} r_2(A_2))^* \rightarrow (r_1(B) \approx_{\mathcal{D}} r_2(B))\big) = \\
&\bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} S(A_1, A_2)^* \rightarrow \big((r_1(A_2) \approx_{\mathcal{D}} r_2(A_2))^* \rightarrow (r_1(B) \approx_{\mathcal{D}} r_2(B))\big) = \\
&\bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} \big(S(A_1, A_2)^* \otimes (r_1(A_2) \approx_{\mathcal{D}} r_2(A_2))^*\big) \rightarrow (r_1(B) \approx_{\mathcal{D}} r_2(B)).
\end{aligned}$$

Using the fact that in every residuated lattice with hedge $a^* \otimes b^* \leq (a \otimes b)^*$ is true for any $a, b \in L$, together with (5.13) and monotony of * we have:

$$\begin{aligned}
&S(A_1, A_2)^* \otimes (r_1(A_2) \approx_{\mathcal{D}} r_2(A_2))^* \leq \\
&\big(S(A_1, A_2) \otimes (r_1(A_2) \approx_{\mathcal{D}} r_2(A_2))\big)^* \leq \\
&(r_1(A_1) \approx_{\mathcal{D}} r_2(A_1))^*.
\end{aligned}$$

Using the previous observation together with antitony of $\rightarrow$ in the first argument:

$$\begin{aligned}
&S(A_1, A_2)^* \rightarrow ||A_2 \Rightarrow B||_{\mathcal{D}} = \\
&\bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} \big(S(A_1, A_2)^* \otimes (r_1(A_2) \approx_{\mathcal{D}} r_2(A_2))^*\big) \rightarrow (r_1(B) \approx_{\mathcal{D}} r_2(B)) \geq \\
&\bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} (r_1(A_1) \approx_{\mathcal{D}} r_2(A_1))^* \rightarrow (r_1(B) \approx_{\mathcal{D}} r_2(B)) = ||A_1 \Rightarrow B||_{\mathcal{D}}.
\end{aligned}$$

□

To sum up, we obtain the following Theorem.

Theorem 48. *Let $A_1, A_2, B_1, B_2 \in \mathbf{L}^R$. For any RDT $\mathcal{D}$ on R and for fixed hedge $*$ we have:*

$$S(A_1, A_2)^* \otimes S(B_2, B_1) \otimes ||A_1 \Rightarrow B_1||_{\mathcal{D}} \leq ||A_2 \Rightarrow B_2||_{\mathcal{D}}, \tag{5.16}$$

$$E(A_1, A_2)^* \otimes E(B_2, B_1) \leq ||A_1 \Rightarrow B_1||_{\mathcal{D}} \leftrightarrow ||A_2 \Rightarrow B_2||_{\mathcal{D}}. \tag{5.17}$$

Proof. The inequality (5.16) is a direct consequence of Lemma 46 and Lemma 47. Equation (5.17) is a consequence of (5.16) and can be proved following the same arguments as in the prove of (5.7):

$$\begin{aligned}
&E(A_1, A_2)^* \otimes E(B_2, B_1) = \\
&(S(A_1, A_2) \wedge S(A_2, A_1))^* \otimes (S(B_2, B_1) \wedge S(B_1, B_2)) \leq \\
&(S(A_1, A_2)^* \wedge S(A_2, A_1)^*) \otimes (S(B_2, B_1) \wedge S(B_1, B_2)) \leq
\end{aligned}$$

$$(S(A_1, A_2)^* \otimes S(B_2, B_1)) \wedge (S(A_2, A_1)^* \otimes S(B_1, B_2)) \leq$$
$$(||A_1 \Rightarrow B_1||_{\mathcal{D}} \rightarrow ||A_2 \Rightarrow B_2||_{\mathcal{D}}) \wedge (||A_2 \Rightarrow B_2||_{\mathcal{D}} \rightarrow ||A_1 \Rightarrow B_1||_{\mathcal{D}}) =$$
$$||A_1 \Rightarrow B_1||_{\mathcal{D}} \leftrightarrow ||A_2 \Rightarrow B_2||_{\mathcal{D}}.$$

□

5.3 Conclusions

In this chapter we have introduced a similarity measure for RDTs (called rank-based similarity) and presented several estimates for similarity-based functional dependencies. Future research will focus on similarity estimates for other similarity measures of RDTs, for example for the tuple-based similarity of RDTs from Chapter 6. We are also going to study weather such similarity estimates are possible for other generalizations of FD from Chapter 3.

Chapter 6

Similarity estimates of query results

In this chapter we will show that relational operations from Section 2.3.2 are robust because they are *insensitive to slight changes in data*: (very) similar input data cannot yield (very) different results under the notions of similarity defined by (5.1). This has many practical implications. For instance, if two experts are asked to assign ranks in a datatable based on their knowledge of particular problem domain, they can come up with different ranks. If the assigned ranks are sufficiently close, we know that we can take either of the ranked data tables and it will produce similar results as the other one when used in subsequent queries. Later in this chapter we will provide an alternative measure of similarity of RDTs based on ranks and tuple values, and we will introduce related relational operation—a similarity-based closure.

6.1 Similarity estimates for relational operations

Before we will show how various relational operations preserve subsethood and similarity degrees, we prove a technical lemma which allows us to derive observations of preserving similarity (5.1) based on observations on preserving subsethood degrees (5.2). The lemma can be used to draw general conclusions about n-ary operations on RDTs, i.e., operations f which map input RDTs $\mathcal{D}_1, \ldots, \mathcal{D}_n$ to $f(\mathcal{D}_1, \ldots, \mathcal{D}_n)$.

Lemma 49. *Let f be an n-ary operation on RDTs. If*

$$\bigotimes_{i=1}^{j} S(\mathcal{D}_i, \mathcal{D}'_i) \otimes \bigotimes_{i=j+1}^{n} S(\mathcal{D}'_i, \mathcal{D}_i) \leq S(f(\mathcal{D}_1, \ldots, \mathcal{D}_n), f(\mathcal{D}'_1, \ldots, \mathcal{D}'_n)), \tag{6.1}$$

for some $0 \leq j \leq n$ and all $\mathcal{D}_i, \mathcal{D}'_i$ $(i = 1, \ldots, n)$, then

$$\bigotimes_{i=1}^{n} E(\mathcal{D}_i, \mathcal{D}'_i) \leq E(f(\mathcal{D}_1, \ldots, \mathcal{D}_n), f(\mathcal{D}'_1, \ldots, \mathcal{D}'_n)) \tag{6.2}$$

for all RDTs $\mathcal{D}_i, \mathcal{D}'_i$ where $i = 1, \ldots, n$.

Proof. First, observe that j in (6.1) splits the arguments to f into two subsets: arguments which are isotone with respect to the graded subsethood and antitone with respect to the graded subsethood: If $j = 0$ and $j = n$ then all considered arguments are antitone and isotone, respectively. Anyway, if (6.1) holds for j and any RDTs $\mathcal{D}_i, \mathcal{D}'_i$, we can use (6.1) twice to get

$$\begin{aligned}&\big(\textstyle\bigotimes_{i=1}^{j} S(\mathcal{D}_i, \mathcal{D}'_i) \otimes \bigotimes_{i=j+1}^{n} S(\mathcal{D}'_i, \mathcal{D}_i)\big) \wedge \big(\bigotimes_{i=1}^{j} S(\mathcal{D}'_i, \mathcal{D}_i) \otimes \bigotimes_{i=j+1}^{n} S(\mathcal{D}_i, \mathcal{D}'_i)\big) \leq \\ &S(f(\mathcal{D}_1, \ldots, \mathcal{D}_n), f(\mathcal{D}'_1, \ldots, \mathcal{D}'_n)) \wedge S(f(\mathcal{D}'_1, \ldots, \mathcal{D}'_n), f(\mathcal{D}_1, \ldots, \mathcal{D}_n)) = \\ &E(f(\mathcal{D}_1, \ldots, \mathcal{D}_n), f(\mathcal{D}'_1, \ldots, \mathcal{D}'_n)).\end{aligned}$$

Now, using (5.8) and (5.3)

$$\textstyle\bigotimes_{i=1}^{n} E(\mathcal{D}_i, \mathcal{D}'_i) = \bigotimes_{i=1}^{n} \big(S(\mathcal{D}_i, \mathcal{D}'_i) \wedge S(\mathcal{D}'_i, \mathcal{D}_i)\big) \leq E(f(\mathcal{D}_1, \ldots, \mathcal{D}_n), f(\mathcal{D}'_1, \ldots, \mathcal{D}'_n)),$$

which proves the claim. □

Note that by a particularization of Lemma 49, for a unary operation f, we get that if

$$S(\mathcal{D}, \mathcal{D}') \leq S(f(\mathcal{D}), f(\mathcal{D}')), \tag{6.3}$$

for all RDTs $\mathcal{D}$ and $\mathcal{D}'$, then

$$E(\mathcal{D}, \mathcal{D}') \leq E(f(\mathcal{D}), f(\mathcal{D}')), \tag{6.4}$$

for all RDTs $\mathcal{D}$ and $\mathcal{D}'$. Condition (6.3) can be seen as a stronger form of isotony. Indeed, f being isotone means that $f(\mathcal{D}) \subseteq f(\mathcal{D}')$ whenever $\mathcal{D} \subseteq \mathcal{D}'$. Clearly, if (6.3) holds then $f(\mathcal{D}) \subseteq f(\mathcal{D}')$ is an abbreviation for $S(\mathcal{D}, \mathcal{D}') = 1$ which implies $S(f(\mathcal{D}), f(\mathcal{D}')) = 1$, i.e., $f(\mathcal{D}) \subseteq f(\mathcal{D}')$ and thus f is isotone.

Further subsections describe similarity estimates for relational operations.

6.1.1 Boolean-like operations

The following assertion shows that $\cup$ and $\cap$ preserve subsethood degrees and similarity degrees given by (5.2) and (5.1), respectively. Putting Theorem 50 into words, (6.5) can be read as: the degree to which $\mathcal{D}_1 \cup \mathcal{D}_2$ is included in $\mathcal{D}'_1 \cup \mathcal{D}'_2$ is at least as high as the degree to which $\mathcal{D}_1$ is included in $\mathcal{D}'_1$ and $\mathcal{D}_2$ is included in $\mathcal{D}'_2$. Following the same principle for the Equation (6.7): the degree to which $\mathcal{D}_1 \cup \mathcal{D}_2$ is similar to $\mathcal{D}'_1 \cup \mathcal{D}'_2$ is at least as high as the degree to which $\mathcal{D}_1$ is similar to $\mathcal{D}'_1$ and $\mathcal{D}_2$ is similar $\mathcal{D}'_2$.

Theorem 50. *For any $\mathcal{D}_1$, $\mathcal{D}'_1$, $\mathcal{D}_2$, and $\mathcal{D}'_2$ on relation scheme R,*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2), \tag{6.5}$$

$$S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \cap \mathcal{D}_2, \mathcal{D}'_1 \cap \mathcal{D}'_2), \tag{6.6}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \wedge E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2), \tag{6.7}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \wedge E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \cap \mathcal{D}_2, \mathcal{D}'_1 \cap \mathcal{D}'_2). \tag{6.8}$$

Proof. (6.5): We need to prove that

$$S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2) \leq \bigwedge_{r \in \mathrm{Tupl}(R)} \big((\mathcal{D}_1 \cup \mathcal{D}_2)(r) \to (\mathcal{D}'_1 \cup \mathcal{D}'_2)(r)\big).$$

Using adjointness, it suffices to check that

$$\big(S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2)\big) \otimes (\mathcal{D}_1 \cup \mathcal{D}_2)(r) \leq (\mathcal{D}'_1 \cup \mathcal{D}'_2)(r)$$

is true for any $r \in \mathrm{Tupl}(R)$. Using (5.2), (2.17) and (2.13):

$$\begin{aligned}
&\big(S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2)\big) \otimes (\mathcal{D}_1 \cup \mathcal{D}_2)(r) \leq \\
&\big((\mathcal{D}_1(r) \to \mathcal{D}'_1(r)) \wedge (\mathcal{D}_2(r) \to \mathcal{D}'_2(r))\big) \otimes (\mathcal{D}_1 \cup \mathcal{D}_2)(r) = \\
&\big((\mathcal{D}_1(r) \to \mathcal{D}'_1(r)) \wedge (\mathcal{D}_2(r) \to \mathcal{D}'_2(r))\big) \otimes (\mathcal{D}_1(r) \vee \mathcal{D}_2(r)) = \\
&\big(\big((\mathcal{D}_1(r) \to \mathcal{D}'_1(r)) \wedge (\mathcal{D}_2(r) \to \mathcal{D}'_2(r))\big) \otimes \mathcal{D}_1(t)\big) \vee \\
&\big(\big((\mathcal{D}_1(r) \to \mathcal{D}'_1(r)) \wedge (\mathcal{D}_2(r) \to \mathcal{D}'_2(r))\big) \otimes \mathcal{D}_2(r)\big) \leq \\
&\big((\mathcal{D}_1(r) \to \mathcal{D}'_1(r)) \otimes \mathcal{D}_1(r)\big) \vee \big((\mathcal{D}_2(r) \to \mathcal{D}'_2(r)) \otimes \mathcal{D}_2(r)\big) \\
&\mathcal{D}'_1(r) \vee \big((\mathcal{D}_2(r) \to \mathcal{D}'_2(r)) \otimes \mathcal{D}_2(r)\big) \leq \\
&\mathcal{D}'_1(r) \vee \mathcal{D}'_2(r) = (\mathcal{D}'_1 \cup \mathcal{D}'_2)(r).
\end{aligned}$$

(6.6): Using the same idea, in order to prove (6.6) it is sufficient to check that for any $r \in \mathrm{Tupl}(R)$

$$\big(S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2)\big) \otimes (\mathcal{D}_1 \cap \mathcal{D}_2)(r) \leq (\mathcal{D}'_1 \cap \mathcal{D}'_2)(r).$$

Using the monotony of $\otimes$ and $\wedge$, and (2.21) we obtain:

$$\begin{aligned}
&\big(S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2)\big) \otimes (\mathcal{D}_1 \cap \mathcal{D}_2)(r) \leq \\
&\big((\mathcal{D}_1(r) \to \mathcal{D}'_1(r)) \wedge (\mathcal{D}_2(r) \to \mathcal{D}'_2(r))\big) \otimes (\mathcal{D}_1 \cap \mathcal{D}_2)(r) = \\
&\big((\mathcal{D}_1(r) \to \mathcal{D}'_1(r)) \wedge (\mathcal{D}_2(r) \to \mathcal{D}'_2(r))\big) \otimes (\mathcal{D}_1(r) \wedge \mathcal{D}_2(r)) \leq \\
&\big((\mathcal{D}_1(r) \to \mathcal{D}'_1(r)) \otimes (\mathcal{D}_1(r) \wedge \mathcal{D}_2(r))\big) \wedge \big((\mathcal{D}_2(r) \to \mathcal{D}'_2(r)) \otimes (\mathcal{D}_1(r) \wedge \mathcal{D}_2(r))\big) \leq \\
&\big((\mathcal{D}_1(r) \to \mathcal{D}'_1(r)) \otimes \mathcal{D}_1(r)\big) \wedge \big((\mathcal{D}_2(r) \to \mathcal{D}'_2(r)) \otimes \mathcal{D}_2(r)\big) \leq \\
&\mathcal{D}'_1(r) \wedge \mathcal{D}'_2(r) = (\mathcal{D}'_1 \cap \mathcal{D}'_2)(r).
\end{aligned}$$

The proof of (6.7) is straightforward. By applying (6.5) twice:

$$\begin{aligned}
&E(\mathcal{D}_1, \mathcal{D}'_1) \wedge E(\mathcal{D}_2, \mathcal{D}'_2) = S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}'_1, \mathcal{D}_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2) \wedge S(\mathcal{D}'_2, \mathcal{D}_2) \leq \\
&S(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2) \wedge S(\mathcal{D}'_1 \cup \mathcal{D}'_2, \mathcal{D}_1 \cup \mathcal{D}_2) = E(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2).
\end{aligned}$$

(6.8) can be proved in the same way as (6.7). □

Optimality

All (6.5)–(6.8) are optimal in the following sense: For any $a \in L$ there are $\mathcal{D}_1$, $\mathcal{D}'_1$, $\mathcal{D}_2$, $\mathcal{D}'_2$ such that

$$S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2) = S(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2) = a$$

and similarly for (6.6)–(6.8). Therefore, one cannot find an expression which can be substituted for the left-hand side of (6.5)–(6.8) and which produces strictly higher similarity estimates for all RDTs. Indeed, in the case of (6.5) and (6.7): for any $a \in L$ consider $\mathcal{D}_1 = \mathcal{D}'_1 = 0_\emptyset$, $\mathcal{D}_2 = 1_\emptyset$ and $\mathcal{D}'_2 = a_\emptyset$. Then

$$\begin{aligned}
&S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2) = (0 \to 0) \wedge (1 \to a) = 1 \wedge a = a,\\
&S(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2) = (0 \vee 1) \to (0 \vee a) = 1 \to a = a,\\
&E(\mathcal{D}_1, \mathcal{D}'_1) \wedge E(\mathcal{D}_2, \mathcal{D}'_2) = 1 \wedge a = a,\\
&E(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2) = 1 \leftrightarrow a = a.
\end{aligned}$$

The optimality of (6.6) and (6.8) can be seen by taking $\mathcal{D}_1 = \mathcal{D}'_1 = \mathcal{D}_2 = 1_\emptyset$ and $\mathcal{D}'_2 = a_\emptyset$ for any $a \in L$. Analogous observation can be made for other estimates in this section.

The estimates we will investigate further in this section employ $\otimes$ instead of $\wedge$ for combining subsethood degrees. Since $\wedge$ is an upper bound for $\otimes$ in $\mathbf{L}$ we have

$$S(\mathcal{D}_1, \mathcal{D}_2) \otimes S(\mathcal{D}'_1, \mathcal{D}'_2) \leq S(\mathcal{D}_1, \mathcal{D}_2) \wedge S(\mathcal{D}'_1, \mathcal{D}'_2)$$

and analogously for E. Therefore the Corollary 51 immediately follows from Theorem 50.

Corollary 51. *For any $\mathcal{D}_1$, $\mathcal{D}'_1$, $\mathcal{D}_2$, and $\mathcal{D}'_2$ on relation scheme R,*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2), \tag{6.9}$$

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \cap \mathcal{D}_2, \mathcal{D}'_1 \cap \mathcal{D}'_2), \tag{6.10}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2), \tag{6.11}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \cap \mathcal{D}_2, \mathcal{D}'_1 \cap \mathcal{D}'_2). \tag{6.12}$$

Estimates given by Corollary 51 are optimal as well: for any estimate from Corollary 51 one can not find another estimate which will be strictly higher for *all* RDTs. Although there exists estimates which are strictly higher for *some* RDT (e.g. the estimates from Theorem 50).

Remark 20. *Let us note that inclusion estimates like those from Theorem 50 do not have a nontrivial interpretation in the original Codd's model of data. For instance, if* $\mathbf{L}$ *is the two-element Boolean algebra, the left-hand side of* (6.5) *is either* 0 *or* 1*. Clearly, $S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2) = 1$ iff $\mathcal{D}_1$ is a subset of $\mathcal{D}'_1$ (in the usual sense) and $\mathcal{D}_2$ is a subset of $\mathcal{D}'_2$, from which one immediately derives that $\mathcal{D}_1 \cup \mathcal{D}_2$ is a subset of $\mathcal{D}'_1 \cup \mathcal{D}'_2$. A similar situation applies for* (6.6)–(6.12).

6.1.2 Ternary residuum

The next operation we consider is the ternary residuum $\twoheadrightarrow$, see (2.50), which can be seen as a ternary counterpart of $\to$ with one of the argument serving as a range. Note that the lower estimate (6.13) differs from the previous inequalities (6.5)–(6.12) in the sense that we use $S(\mathcal{D}'_1, \mathcal{D}_1)$ and not $S(\mathcal{D}_1, \mathcal{D}'_1)$. This is a consequence of the antitony of $\to$ in the first argument, see (2.6).

Theorem 52. *For any $\mathcal{D}_1$, $\mathcal{D}'_1$, $\mathcal{D}_2$, $\mathcal{D}'_2$, $\mathcal{D}_3$, and $\mathcal{D}'_3$ on R, we have:*

$$S(\mathcal{D}'_1, \mathcal{D}_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \leq S(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2, \mathcal{D}'_1 \twoheadrightarrow^{\mathcal{D}'_3} \mathcal{D}'_2), \tag{6.13}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \otimes E(\mathcal{D}_3, \mathcal{D}'_3) \leq E(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2, \mathcal{D}'_1 \twoheadrightarrow^{\mathcal{D}'_3} \mathcal{D}'_2). \tag{6.14}$$

Proof. In order to prove (6.13) we need to show that the following inequality is true for each $r \in \mathrm{Tupl}(R)$:

$$S(\mathcal{D}'_1, \mathcal{D}_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \otimes (\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2)(r) \leq (\mathcal{D}'_1 \twoheadrightarrow^{\mathcal{D}'_3} \mathcal{D}'_2)(r).$$

The claim is a consequence of (2.13), (2.16) and monotony of $\otimes$.

$$\begin{aligned}
&S(\mathcal{D}'_1, \mathcal{D}_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \otimes (\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2)(r) = \\
&S(\mathcal{D}'_1, \mathcal{D}_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \otimes \mathcal{D}_3(r) \otimes \big(\mathcal{D}_1(r) \to \mathcal{D}_2(r)\big) \leq \\
&S(\mathcal{D}'_1, \mathcal{D}_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes \mathcal{D}_3(r) \otimes \big(\mathcal{D}_3(r) \to \mathcal{D}'_3(r)\big) \otimes \big(\mathcal{D}_1(r) \to \mathcal{D}_2(r)\big) \leq \\
&S(\mathcal{D}'_1, \mathcal{D}_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes \mathcal{D}'_3(r) \otimes \big(\mathcal{D}_1(r) \to \mathcal{D}_2(r)\big) \leq \\
&\mathcal{D}'_3(r) \otimes \big(\mathcal{D}'_1(r) \to \mathcal{D}_1(r)\big) \otimes \big(\mathcal{D}_1(r) \to \mathcal{D}_2(r)\big) \otimes \big(\mathcal{D}_2(r) \to \mathcal{D}'_2(r)\big) \leq \\
&\mathcal{D}'_3(r) \otimes \big(\mathcal{D}'_1(r) \to \mathcal{D}_2(r)\big) \otimes \big(\mathcal{D}_2(r) \to \mathcal{D}'_2(r)\big) \leq \\
&\mathcal{D}'_3(r) \otimes \big(\mathcal{D}'_1(r) \to \mathcal{D}'_2(r)\big) = (\mathcal{D}'_1 \twoheadrightarrow^{\mathcal{D}'_3} \mathcal{D}'_2)(r).
\end{aligned}$$

The inequality (6.14) is a consequence of Lemma 49 and Equation (6.13). □

Corollary 53. *For any $\mathcal{D}_1, \mathcal{D}'_1, \mathcal{D}_2, \mathcal{D}'_2$ on R:*

$$S(\mathcal{D}'_1, \mathcal{D}_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c \to c') \leq S(\mathcal{D}_2 \boxbslash_c \mathcal{D}_1, \mathcal{D}'_2 \boxbslash_{c'} \mathcal{D}'_1), \tag{6.15}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c \leftrightarrow c') \leq E(\mathcal{D}_2 \boxbslash_c \mathcal{D}_1, \mathcal{D}'_2 \boxbslash_{c'} \mathcal{D}'_1), \tag{6.16}$$

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c' \to c) \leq S(c \twoheadrightarrow^{\mathcal{D}_2} \mathcal{D}_1, c' \twoheadrightarrow^{\mathcal{D}'_2} \mathcal{D}'_1), \tag{6.17}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c \leftrightarrow c') \leq E(c \twoheadrightarrow^{\mathcal{D}_2} \mathcal{D}_1, c' \twoheadrightarrow^{\mathcal{D}'_2} \mathcal{D}'_1). \tag{6.18}$$

Proof. Inequalities (6.15) and (6.16) follow from Theorem 52 and (2.51). Furthermore, (6.17) and (6.18) follow from Theorem 52 and (2.52). □

6.1.3 Projection and division

We will now establish similarity estimates for projection and residuated division. We start by considering projection (2.53):

Theorem 54. *Let $\mathcal{D}$ and $\mathcal{D}'$ be RDTs on relation scheme R_1 and let $R_2 \subseteq R_1$. Then*

$$S(\mathcal{D}, \mathcal{D}') \leq S(\pi_{R_2}(\mathcal{D}), \pi_{R_2}(\mathcal{D}')), \tag{6.19}$$

$$E(\mathcal{D}, \mathcal{D}') \leq E(\pi_{R_2}(\mathcal{D}), \pi_{R_2}(\mathcal{D}')). \tag{6.20}$$

Proof. We prove the first inequality (6.19), the second one is a consequence of Lemma 49. Using adjointness, we need to check that

$$S(\mathcal{D}, \mathcal{D}') \otimes (\pi_{R_2}(\mathcal{D}))(r_2) \leq (\pi_{R_2}(\mathcal{D}'))(r_2)$$

holds true for any $r_2 \in \mathrm{Tupl}(R_2)$. By definition of $\pi_{R_2}(\mathcal{D})$ and using the fact that $\otimes$ is distributive over $\bigvee$ (2.17) we get

$$\begin{aligned}
&S(\mathcal{D},\mathcal{D}') \otimes (\pi_{R_2}(\mathcal{D}))(r_2) = \\
&S(\mathcal{D},\mathcal{D}') \otimes \textstyle\bigvee_{r_3 \in \mathrm{Tupl}(R_3)} \mathcal{D}(r_2 r_3) = \\
&\textstyle\bigvee_{r_3 \in \mathrm{Tupl}(R_3)} \big(S(\mathcal{D},\mathcal{D}') \otimes \mathcal{D}(r_2 r_3)\big),
\end{aligned}$$

where $R_3 = R_1 \setminus R_2$. From the fact that $r_2 r_3 \in \mathrm{Tupl}(R_1)$, we further obtain:

$$\begin{aligned}
&\textstyle\bigvee_{r_3 \in \mathrm{Tupl}(R_3)} \big(S(\mathcal{D},\mathcal{D}') \otimes \mathcal{D}(r_2 r_3)\big) \leq \\
&\textstyle\bigvee_{r_3 \in \mathrm{Tupl}(R_3)} \big((\mathcal{D}(r_2 r_3) \to \mathcal{D}'(r_2 r_3)) \otimes \mathcal{D}(r_2 r_3)\big) \leq \\
&\textstyle\bigvee_{r_3 \in \mathrm{Tupl}(R_3)} \mathcal{D}'(r_2 r_3) = (\pi_{R_2}(\mathcal{D}'))(r_2).
\end{aligned}$$

□

Now we turn our attention to residuated division (2.54), which was also introduced in the Section 2.3. We repeat the definition here mainly for convenience. Let $\mathcal{D}_1$ be an RDT on R_1, let $\mathcal{D}_2$ be an RDT on $R_2 \subseteq R_1$, and let $\mathcal{D}_3$ be an RDT on $R_3 = R_1 \setminus R_2$. Then, a *division* of $\mathcal{D}_1$ by $\mathcal{D}_2$ which ranges over $\mathcal{D}_3$ is an RDT on R_3:

$$\begin{aligned}
\big(\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2\big)(r_3) &= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_2(r_2) \twoheadrightarrow^{\mathcal{D}_3(r_3)} \mathcal{D}_1(r_2 r_3)\big) \\
&= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_3(r_3) \otimes (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2 r_3))\big). \qquad (6.21)
\end{aligned}$$

First, let us note that residuated division can be used to express containment and similarity degrees of RDTs. Consider the borderline case of residuated division when $R_1 = R_2$ (and thus $R_3 = \emptyset$):

$$\begin{aligned}
\big(\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2\big)(\emptyset) &= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_2(r_2) \twoheadrightarrow^{\mathcal{D}_3(\emptyset)} \mathcal{D}_1(r_2 \emptyset)\big) \\
&= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_3(\emptyset) \otimes (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2))\big).
\end{aligned}$$

Since $R_3 = \emptyset$, $\mathcal{D}_3$ is an RDT on empty relational scheme. In particular, by choosing $\mathcal{D}_3 = 1_\emptyset$ we obtain:

$$\begin{aligned}
\big(\mathcal{D}_1 \div^{1} \mathcal{D}_2\big)(\emptyset) &= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(1 \otimes (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2))\big) \\
&= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2)) = S(\mathcal{D}_2,\mathcal{D}_1).
\end{aligned}$$

As a consequence, (5.2) and (5.1) are expressible inside the model of Belohlavek and Vychodil and thus the similarity estimations are relational *per se*. The only conceptual difference between (5.2) and $\big(\mathcal{D}_1 \div^{1} \mathcal{D}_2\big)(\emptyset)$ is that the value of (5.2) is a degree from L whereas the $\big(\mathcal{D}_1 \div^{1} \mathcal{D}_2\big)(\emptyset)$ is an RDT on the empty relation scheme which represents the degree. The case of (5.1) is analogous.

The similarity estimates for residuated division are described by the following theorem.

Theorem 55. *Let $\mathcal{D}_1$, $\mathcal{D}'_1$ be RDTs on R_1, $\mathcal{D}_2$, $\mathcal{D}'_2$ be RDTs on $R_2 \subseteq R_1$, and $\mathcal{D}_3$, $\mathcal{D}'_3$ be RDTs on $R_3 = R_1 \setminus R_2$, respectively. Then*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}'_2, \mathcal{D}_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \leq S(\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2, \mathcal{D}'_1 \div^{\mathcal{D}'_3} \mathcal{D}'_2), \tag{6.22}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \otimes E(\mathcal{D}_3, \mathcal{D}'_3) \leq E(\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2, \mathcal{D}'_1 \div^{\mathcal{D}'_3} \mathcal{D}'_2). \tag{6.23}$$

Proof. For proving (6.22), we need to verify that

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}'_2, \mathcal{D}_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \otimes (\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2)(r_3) \leq (\mathcal{D}'_1 \div^{\mathcal{D}'_3} \mathcal{D}'_2)(r_3)$$

for every $r_3 \in \mathrm{Tupl}(R_3)$. By using (2.21) and (2.16) we obtain:

$$\begin{aligned}
&S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}'_2, \mathcal{D}_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \otimes (\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2)(r_3) = \\
&S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}'_2, \mathcal{D}_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \otimes \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_3(r_3) \otimes (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2 r_3))\big) \leq \\
&\textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}'_2, \mathcal{D}_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \otimes \mathcal{D}_3(r_3) \otimes (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2 r_3))\big).
\end{aligned}$$

Using the same ideas as in the proof of Theorem 52 we observe:

$$\begin{aligned}
&\textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}'_2, \mathcal{D}_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \otimes \mathcal{D}_3(r_3) \otimes (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2 r_3))\big) \leq \\
&\textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}'_3(r_3) \otimes (\mathcal{D}'_2(r_2) \to \mathcal{D}'_1(r_2 r_3))\big) = (\mathcal{D}'_1 \div^{\mathcal{D}'_3} \mathcal{D}'_2)(r_3).
\end{aligned}$$

The second inequality (6.23) follows from (6.22) by applying Lemma 49. □

6.1.4 Similarity-based restriction

The basic characterization of similarity estimates for similarity-based restriction (2.55) is the following.

Theorem 56. *Let $\mathcal{D}$ and $\mathcal{D}'$ be RDTs on relation scheme R and let $y \in R$ and $d \in D_y$. Then,*

$$S(\mathcal{D}, \mathcal{D}') \leq S(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx d}(\mathcal{D}')), \tag{6.24}$$

$$E(\mathcal{D}, \mathcal{D}') \leq E(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx d}(\mathcal{D}')). \tag{6.25}$$

Proof. Again, using the adjointness of $\otimes$ and $\to$, in order to prove (6.24), it suffices to check that

$$S(\mathcal{D}, \mathcal{D}') \otimes (\sigma_{y \approx d}(\mathcal{D}))(r) \leq (\sigma_{y \approx d}(\mathcal{D}'))(r)$$

holds for all tuples $r \in \mathrm{Tupl}(R)$. Using the definition of $\sigma_{y \approx d}(\mathcal{D})$ and (2.13), we observe that:

$$\begin{aligned}
&S(\mathcal{D}, \mathcal{D}') \otimes (\sigma_{y \approx d}(\mathcal{D}))(r) = \\
&S(\mathcal{D}, \mathcal{D}') \otimes \mathcal{D}(r) \otimes r(y) \approx_y d \leq \\
&(\mathcal{D}(r) \to \mathcal{D}'(r)) \otimes \mathcal{D}(r) \otimes r(y) \approx_y d \leq \\
&\mathcal{D}'(r) \otimes r(y) \approx_y d = \sigma_{y \approx d}(\mathcal{D}').
\end{aligned}$$

Since $r \in \mathrm{Tupl}(R)$ has been taken arbitrarily, we can conclude that (6.24) holds true. Now, the inequality (6.25) is a consequence of Lemma 49. □

Remark 21. *We have proved the similarity estimates for a similarity-based condition $y \approx d$. The estimates given by Theorem 56 will remain valid for general comparators [82].*

The similarity estimates in Theorem 56 involve two restrictions using the same constant d from the domain of y. Intuitively, we may expect that two restrictions that use different constants d and d' should yield similar results if d and d' are similar. This can be shown if the similarity on the domain of y is $\otimes$-transitive.

Theorem 57. *Let $\mathcal{D}$ be an RDT on R, let $y \in R$, $d, d' \in D_y$, and let $\approx_y$ be $\otimes$-transitive. Then*

$$d \approx_y d' \leq S(\sigma_{y\approx d}(\mathcal{D}), \sigma_{y\approx d'}(\mathcal{D})). \tag{6.26}$$

Proof. By adjointness, for verifying (6.26) it is sufficient to check that

$$d \approx_y d' \otimes (\sigma_{y\approx d}(\mathcal{D}))(r) \leq (\sigma_{y\approx d'}(\mathcal{D}))(r)$$

is true for all $r \in \mathrm{Tupl}(R)$. Using the $\otimes$-transitivity of $\approx_y$ and isotony of $\otimes$, for each $r \in \mathrm{Tupl}(R)$ we have:

$$\begin{aligned}
&d \approx_y d' \otimes (\sigma_{y\approx d}(\mathcal{D}))(r) = \\
&d \approx_y d' \otimes (\mathcal{D}(r) \otimes r(y) \approx_y d) = \\
&\mathcal{D}(r) \otimes (r(y) \approx_y d \otimes d \approx_y d') \leq \\
&\mathcal{D}(r) \otimes r(y) \approx_y d' = (\sigma_{y\approx d'}(\mathcal{D}))(r).
\end{aligned}$$

□

As a consequence of Theorem 56 and Theorem 57 we obtain the following corollary.

Corollary 58. *Let $\mathcal{D}$ and $\mathcal{D}'$ be RDTs on R and let $y \in R$, $d, d' \in D_y$ and $\approx_y$ be $\otimes$-transitive. Then,*

$$S(\mathcal{D}, \mathcal{D}') \otimes d \approx_y d' \leq S(\sigma_{y\approx d}(\mathcal{D}), \sigma_{y\approx d'}(\mathcal{D}')), \tag{6.27}$$

$$E(\mathcal{D}, \mathcal{D}') \otimes d \approx_y d' \leq E(\sigma_{y\approx d}(\mathcal{D}), \sigma_{y\approx d'}(\mathcal{D}')). \tag{6.28}$$

Proof. Using Theorem 56, Theorem 57 and (2.36)

$$\begin{aligned}
&S(\mathcal{D}, \mathcal{D}') \otimes d \approx_y d' \leq \\
&S(\sigma_{y\approx d}(\mathcal{D}), \sigma_{y\approx d}(\mathcal{D}')) \otimes d \approx_y d' \leq \\
&S(\sigma_{y\approx d}(\mathcal{D}), \sigma_{y\approx d}(\mathcal{D}')) \otimes S(\sigma_{y\approx d}(\mathcal{D}'), \sigma_{y\approx d'}(\mathcal{D}')) \leq \\
&S(\sigma_{y\approx d}(\mathcal{D}), \sigma_{y\approx d'}(\mathcal{D}')),
\end{aligned}$$

proving (6.27). The inequality (6.28) is a consequence of (6.27) and (2.21).

$$\begin{aligned}
&E(\mathcal{D}, \mathcal{D}') \otimes d \approx_y d' = \\
&\big(S(\mathcal{D}, \mathcal{D}') \wedge S(\mathcal{D}', \mathcal{D})\big) \otimes d \approx_y d' \leq
\end{aligned}$$

$$\big(S(\mathcal{D},\mathcal{D}')\otimes d\approx_y d'\big)\wedge\big(S(\mathcal{D}',\mathcal{D})\otimes d\approx_y d'\big)\leq$$
$$S(\sigma_{y\approx d}(\mathcal{D}),\sigma_{y\approx d'}(\mathcal{D}'))\wedge S(\sigma_{y\approx d'}(\mathcal{D}'),\sigma_{y\approx d}(\mathcal{D}))=E(\sigma_{y\approx d}(\mathcal{D}),\sigma_{y\approx d'}(\mathcal{D}')).$$

□

Remark 22. *Note that Lemma 57 and Corollary 58 do not hold for similarities which are not $\otimes$-transitive. For instance, take $\mathcal{D}$ on $\{y\}$ such that $\mathcal{D}(r)=1$ for r with $r(y)=c$, and $\mathcal{D}(r')=0$ for all $r'\neq r$. Suppose that $\mathbf{L}$ is a complete residuated lattice on the real unit interval with $\otimes$ and $\to$ being the Łukasiewicz operations. Furthermore, suppose that for $d,d'\in D_y$ we have $d\approx_y d'=0.9$, $c\approx_y d=0.8$, and $c\approx_y d'=0$. Obviously, $\approx_y$ is not $\otimes$-transitive since $c\approx_y d\otimes d\approx_y d'=0.8\otimes 0.9=0.7\not\leq 0=c\approx_y d'$. As a consequence, $S(\sigma_{y\approx d}(\mathcal{D}),\sigma_{y\approx d'}(\mathcal{D}))=0.8\to 0=0.2\not\geq 0.9$.*

Further question related to similarity is whether a small change of the definition of domain similarities yields a small change of query results. This type of similarity preservation can also be established in the model. The issue of sensitivity to changes in domain similarities is actually important from the application viewpoint—several different similarities can be defined on a domain and database users may be interested in assessing the impact of using a chosen similarity with respect to other similarity on the domain. The fact that small differences in similarities yield small differences in results shows that the model is robust.

In order to measure containment and similarity of domain similarities, we introduce the following notation. If $\approx_y$ and $\approx'_y$ are similarities on the same domain D_y, we put:

$$S(\approx_y,\approx'_y)=\textstyle\bigwedge_{d_1,d_2\in D_y}\big(d_1\approx_y d_2\to d_1\approx'_y d_2\big),\tag{6.29}$$
$$E(\approx_y,\approx'_y)=\textstyle\bigwedge_{d_1,d_2\in D_y}\big(d_1\approx_y d_2\leftrightarrow d_1\approx'_y d_2\big).\tag{6.30}$$

Notice that (6.29) and (6.30) are defined in a similar way as (5.2) and (5.1) only with domain similarities $\approx_y$ and $\approx'_y$ instead of RDTs.

Considering similarities $\approx_y$ and $\approx'_y$ on the domain of the attribute y, we denote by $\sigma_{y\approx d}(\mathcal{D})$ and $\sigma_{y\approx' d}(\mathcal{D})$ the restrictions which use $\approx_y$ and $\approx'_y$, respectively. Under this notation, we have the following observation:

Theorem 59. *Let $\mathcal{D}$ be RDT on R, $y\in R$, and $d\in D_y$. Furthermore, let $\approx_y$ and $\approx'_y$ be similarities on D_y. Then*

$$S(\mathcal{D},\mathcal{D}')\otimes S(\approx_y,\approx'_y)\leq S(\sigma_{y\approx d}(\mathcal{D}),\sigma_{y\approx' d}(\mathcal{D}')),\tag{6.31}$$
$$E(\mathcal{D},\mathcal{D}')\otimes E(\approx_y,\approx'_y)\leq E(\sigma_{y\approx d}(\mathcal{D}),\sigma_{y\approx' d}(\mathcal{D}')).\tag{6.32}$$

Proof. Analogously as in previous observations, for (6.31) it is sufficient to prove that for each $r\in\mathrm{Tupl}(R)$:

$$S(\mathcal{D},\mathcal{D}')\otimes S(\approx_y,\approx'_y)\otimes(\sigma_{y\approx d}(\mathcal{D}))(r)\leq(\sigma_{y\approx' d}(\mathcal{D}'))(r).$$

Using isotony of $\otimes$ and $\bigwedge$ and utilizing (2.13) twice:

$$\begin{aligned}
&S(\mathcal{D},\mathcal{D}') \otimes S(\approx_y, \approx'_y) \otimes (\sigma_{y\approx d}(\mathcal{D}))(r) = \\
&S(\mathcal{D},\mathcal{D}') \otimes S(\approx_y, \approx'_y) \otimes \mathcal{D}(r) \otimes r(y) \approx_y d \leq \\
&S(\mathcal{D},\mathcal{D}') \otimes (r(y) \approx_y d \to r(y) \approx'_y d) \otimes \mathcal{D}(r) \otimes r(y) \approx_y d = \\
&S(\mathcal{D},\mathcal{D}') \otimes \mathcal{D}(r) \otimes (r(y) \approx_y d) \otimes (r(y) \approx_y d \to r(y) \approx'_y d) \leq \\
&S(\mathcal{D},\mathcal{D}') \otimes \mathcal{D}(r) \otimes r(y) \approx'_y d \leq \\
&\mathcal{D}(r) \otimes (\mathcal{D}(r) \to \mathcal{D}'(r)) \otimes r(y) \approx'_y d \leq \\
&\mathcal{D}'(r) \otimes r(y) \approx'_y d = (\sigma_{y\approx' d}(\mathcal{D}'))(r).
\end{aligned}$$

The inequality (6.32) can be derived from (6.31). □

As a consequence we obtain the following corollary.

Corollary 60. *Let $\mathcal{D}$ and $\mathcal{D}'$ be RDTs on R, let $y \in R$, and let $\approx_y$ and $\approx'_y$ be similarities on D_y. Then,*

$$S(\approx_y, \approx'_y) \leq S(\sigma_{y\approx d}(\mathcal{D}), \sigma_{y\approx' d}(\mathcal{D})), \tag{6.33}$$

$$E(\approx_y, \approx'_y) \leq E(\sigma_{y\approx d}(\mathcal{D}), \sigma_{y\approx' d}(\mathcal{D})). \tag{6.34}$$

Remark 23. *Moreover, the previous assertion can be combined with Corollary 58 to incorporate restrictions using different values from D_y but we do not present the observation here because it can be derived from our general result in Section 6.2.*

6.1.5 Natural and similarity-based joins

Natural joins and their variants have been introduced in Section 2.3. As we have seen, the (equality-based) natural join (2.57) can be considered as the fundamental one, meaning that the other possible joins result from the fundamental one and other relational operations. This enables us to simplify observations of similarity estimates. We will first explore the similarity preservation for the equality-based natural join, and utilize observations on similarity preservation of other operations to get estimates for other joins.

Theorem 61. *Let $\mathcal{D}_1$, $\mathcal{D}'_1$ be RDTs on $R_1 \cup R_3$ and $\mathcal{D}_2$, $\mathcal{D}'_2$ be RDTs on $R_2 \cup R_3$ such that $R_1 \cap R_2 = R_1 \cap R_3 = R_2 \cap R_3 = \emptyset$. Then*

$$S(\mathcal{D}_1,\mathcal{D}'_1) \otimes S(\mathcal{D}_2,\mathcal{D}'_2) \leq S(\mathcal{D}_1 \bowtie \mathcal{D}_2, \mathcal{D}'_1 \bowtie \mathcal{D}'_2), \tag{6.35}$$

$$E(\mathcal{D}_1,\mathcal{D}'_1) \otimes E(\mathcal{D}_2,\mathcal{D}'_2) \leq E(\mathcal{D}_1 \bowtie \mathcal{D}_2, \mathcal{D}'_1 \bowtie \mathcal{D}'_2). \tag{6.36}$$

Proof. Again, in order to prove (6.35) we need to show that

$$S(\mathcal{D}_1,\mathcal{D}'_1) \otimes S(\mathcal{D}_2,\mathcal{D}'_2) \otimes (\mathcal{D}_1 \bowtie \mathcal{D}_2)(r_1r_2r_3) \leq (\mathcal{D}'_1 \bowtie \mathcal{D}'_2)(r_1r_2r_3)$$

is true for any $r_1 \in \mathrm{Tupl}(R_1)$, $r_2 \in \mathrm{Tupl}(R_2)$, and $r_3 \in \mathrm{Tupl}(R_3)$. Using (2.57), (2.13), isotony of $\otimes$ and $\bigwedge$, we have

$$S(\mathcal{D}_1,\mathcal{D}'_1) \otimes S(\mathcal{D}_2,\mathcal{D}'_2) \otimes (\mathcal{D}_1 \bowtie \mathcal{D}_2)(r_1r_2r_3) =$$

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes \mathcal{D}_1(r_1 r_3) \otimes \mathcal{D}_2(r_2 r_3) \leq$$
$$\mathcal{D}_1(r_1 r_3) \otimes (\mathcal{D}_1(r_1 r_3) \to \mathcal{D}'_1(r_1 r_3)) \otimes \mathcal{D}_2(r_2 r_3) \otimes (\mathcal{D}_2(r_2 r_3) \to \mathcal{D}'_2(r_2 r_3)) \leq$$
$$\mathcal{D}'_1(r_1 r_3) \otimes \mathcal{D}'_2(r_2 r_3) = (\mathcal{D}'_1 \bowtie \mathcal{D}'_2)(r_1 r_2 r_3),$$

which proves (6.35) since r_1, r_2, r_3 have been taken arbitrarily. The second claim follows from the first one using Lemma 49. □

Corollary 62. *For any $\mathcal{D}_1$, $\mathcal{D}'_1$, $\mathcal{D}_2$, and $\mathcal{D}'_2$ on relation scheme R,*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \otimes \mathcal{D}_2, \mathcal{D}'_1 \otimes \mathcal{D}'_2), \tag{6.37}$$
$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \otimes \mathcal{D}_2, \mathcal{D}'_1 \otimes \mathcal{D}'_2). \tag{6.38}$$

Proof. Consequence of the fact that $\otimes$-intersection is a special case of (equality-based) natural join, see Section 2. □

Corollary 63. *For any $\mathcal{D}, \mathcal{D}'$ on R:*

$$S(\mathcal{D}, \mathcal{D}') \otimes (c \to c') \leq S(c \otimes \mathcal{D}, c' \otimes \mathcal{D}'), \tag{6.39}$$
$$E(\mathcal{D}, \mathcal{D}') \otimes (c \leftrightarrow c') \leq E(c \otimes \mathcal{D}, c' \otimes \mathcal{D}'). \tag{6.40}$$

Proof. The inequalities (6.39) and (6.40) follow from Theorem 61 by taking $\mathcal{D}_2 = c_\emptyset$ and $\mathcal{D}'_2 = c'_\emptyset$. □

We now turn our attention to particular joins which can be seen as derived relational operations. For the derived joins, we establish similarity preservation theorems based on our previous observations. For example a similarity-based equijoin is in fact a similarity-based restriction of a cross join of two RDTs (defined on disjoint relation schemes). Unlike (2.55), (2.58) uses a restriction based on a more general comparator $y_1 \approx y_2$, where *both* y_1 and y_2 are attributes (from $R_1 \cup R_2$) but this is just a conservative extension of (2.55), cf. Remark 21.

Corollary 64. *Let $\mathcal{D}_1$ and $\mathcal{D}'_1$ be RDTs on R_1 and let $\mathcal{D}_2$ and $\mathcal{D}'_2$ be RDTs on R_2 such that $R_1 \cap R_2 = \emptyset$. Let $y_1 \in R_1$ and $y_2 \in R_2$ have the same domain with similarity. Then*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \bowtie_{y_1 \approx y_2} \mathcal{D}_2, \mathcal{D}'_1 \bowtie_{\bowtie_{y_1 \approx y_2}} \mathcal{D}'_2), \tag{6.41}$$
$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \bowtie_{y_1 \approx y_2} \mathcal{D}_2, \mathcal{D}'_1 \bowtie_{y_1 \approx y_2} \mathcal{D}'_2). \tag{6.42}$$

Proof. The first inequality (6.41) is a consequence of Theorem 61 and Theorem 56:

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \bowtie \mathcal{D}_2, \mathcal{D}'_1 \bowtie \mathcal{D}'_2) \leq$$
$$S(\sigma_{y_1 \approx y_2}(\mathcal{D}_1 \bowtie \mathcal{D}_2), \sigma_{y_1 \approx y_2}(\mathcal{D}'_1 \bowtie \mathcal{D}'_2)) = S(\mathcal{D}_1 \bowtie_{y_1 \approx y_2} \mathcal{D}_2, \mathcal{D}'_1 \bowtie_{y_1 \approx y_2} \mathcal{D}'_2).$$

The inequality (6.42) follows from (6.41) and Lemma 49. □

Another join-like operation with RDTs introduced in Section 2.3 is a similarity-based equijoin with a threshold (2.59), which allows us to put emphasis on the similarity-based condition. For this particular join, we may want to describe lower similarity estimates based not only on the containment and/or similarity of RDTs but also on similarity of the threshold degrees from $\mathbf{L}$ which appear in the restriction condition. The following similarity estimates can be established:

Theorem 65. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be RDTs on R_1 and R_2 such that $R_1 \cap R_2 = \emptyset$. Let $y_1 \in R_1$ and $y_2 \in R_2$ have the same domain with similarity and let $c, c' \in L$. Then*

$$c' \to c \leq S\big(\mathcal{D}_1 \bowtie_{c/y_1 \approx y_2} \mathcal{D}_2, \mathcal{D}_1 \bowtie_{c'/y_1 \approx y_2} \mathcal{D}_2\big), \tag{6.43}$$

$$c \leftrightarrow c' \leq E\big(\mathcal{D}_1 \bowtie_{c/y_1 \approx y_2} \mathcal{D}_2, \mathcal{D}_1 \bowtie_{c'/y_1 \approx y_2} \mathcal{D}_2\big). \tag{6.44}$$

Proof. Observe that using (2.13)

$$\begin{aligned}
&(c' \to c) \otimes (\mathcal{D}_1 \bowtie_{c/y_1 \approx y_2} \mathcal{D}_2)(r_1 r_2) = \\
&(c' \to c) \otimes \mathcal{D}_1(r_1) \otimes \mathcal{D}_2(r_2) \otimes (c \to r_1(y_1) \approx_{y_1} r_2(y_2)) \leq \\
&\mathcal{D}_1(r_1) \otimes \mathcal{D}_2(r_2) \otimes (c' \to r_1(y_1) \approx_{y_1} r_2(y_2)) = \\
&(\mathcal{D}_1 \bowtie_{c'/y_1 \approx y_2} \mathcal{D}_2)(r_1 r_2).
\end{aligned}$$

The inequality (6.43) follows by adjointness. Now, (6.44) is a consequence of (6.43). □

As a consequence:

Corollary 66. *Let $\mathcal{D}_1$ and $\mathcal{D}'_1$ be RDTs on R_1 and let $\mathcal{D}_2$ and $\mathcal{D}'_2$ be RDTs on R_2 such that $R_1 \cap R_2 = \emptyset$. Let $y_1 \in R_1$ and $y_2 \in R_2$ have the same domain with similarity. Then,*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c' \to c) \leq S(\mathcal{D}_1 \bowtie_{c/y_1 \approx y_2} \mathcal{D}_2, \mathcal{D}'_1 \bowtie_{c'/y_1 \approx y_2} \mathcal{D}'_2), \tag{6.45}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c' \leftrightarrow c) \leq E(\mathcal{D}_1 \bowtie_{c/y_1 \approx y_2} \mathcal{D}_2, \mathcal{D}'_1 \bowtie_{c'/y_1 \approx y_2} \mathcal{D}'_2), \tag{6.46}$$

for all $c, c' \in L$.

Moreover, with smaller values of thresholds, the restriction condition is more relaxed and the degrees (2.59) are higher, meaning

$$S\big(\mathcal{D}_1 \bowtie_{c/y_1 \approx y_2} \mathcal{D}_2, \mathcal{D}_1 \bowtie_{c'/y_1 \approx y_2} \mathcal{D}_2\big) = 1$$

whenever $c' \leq c$, i.e., $\mathcal{D}_1 \bowtie_{c/y_1 \approx y_2} \mathcal{D}_2$ is fully contained in $\mathcal{D}_1 \bowtie_{c'/y_1 \approx y_2} \mathcal{D}_2$. This is a consequence of the antitony of the residuum $\to$ in the first argument and the isotony of the multiplication $\otimes$.

6.1.6 Further operations

So far, we have shown that relational operations introduced in Section 2.3 preserve similarities and as a consequence, pairwise similar arguments to operations yield similar results.

Among the operations we have not considered yet are the operations of renaming, kernel, and support which also belong to the basic operations in the model.

Since the operation of renaming only changes the names of attributes without altering the data table as such (i.e. data as well as ranks stay untouched), the renaming preserves similarity trivially. The similarity of input RDTs is simply the same as the similarity of output RDTs and thus $S(\mathcal{D}, \mathcal{D}') = S(\rho_f(\mathcal{D}), \rho_f(\mathcal{D}'))$ and analogously for E.

As we have seen in Section 2.3, kernel and support are unary operations that produce a nonranked table from an RDT. It is easily seen that by nature, neither the kernel nor the support preserve similarity except for the trivial cases, as the following example demonstrates.

Example 5. *Assume data tables $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ on the same relation scheme R with single tuple r such that $\mathcal{D}_1(r) = 1$, $\mathcal{D}_2(r) = 0.3$, $\mathcal{D}_3(r) = 0$. For the Łukasiewicz structure of truth degrees on $[0, 1]$ we have: $S(\mathcal{D}_1, \mathcal{D}_2) = 1 \to 0.3 = 0.3$ and $S(\Delta\mathcal{D}_1, \Delta\mathcal{D}_2) = 1 \to 0 = 0$. Furthermore $S(\mathcal{D}_2, \mathcal{D}_3) = 0.3 \to 0 = 0.7$, whereas $S(\nabla\mathcal{D}_2, \nabla\mathcal{D}_3) = 1 \to 0 = 0$.*

In general, since both $\Delta\mathcal{D}$ and $\nabla\mathcal{D}$ are nonranked, we have $S(\Delta\mathcal{D}, \Delta\mathcal{D}') \in \{0, 1\}$ and $S(\nabla\mathcal{D}, \nabla\mathcal{D}') \in \{0, 1\}$. Thus, the only nontrivial estimations are:

(i) If for each $r \in \mathrm{Tupl}(R)$ such that $\mathcal{D}(r) = 1$,
we have $\mathcal{D}'(r) = 1$, then $S(\Delta\mathcal{D}, \Delta\mathcal{D}') = 1$;

(ii) If for each $r \in \mathrm{Tupl}(R)$ such that $\mathcal{D}(r) > 0$,
we have $\mathcal{D}'(r) > 0$, then $S(\nabla\mathcal{D}, \nabla\mathcal{D}') = 1$;

and analogously for E.

The negative result on preserving similarity by kernel and support should not be interpreted as a weakness of the model. For the majority of queries which are free of kernels and supports, one can utilize all the positive results we have made in this section. In practice, the kernel and supports are used as the "outermost operations", so one can always estimate similarity of the results prior to the application of kernels and supports. In addition to queries which are free of kernels and supports, one may use the nontrivial estimates for queries involving kernels and supports whose arguments are constant (i.e., always the same RDTs). The issues related with compound relational queries are discussed in the next section.

6.2 Similarity of complex query results

The previous section was devoted to the similarity preservation of single relational operations. Usually, relational queries expressed by relational algebra expressions are compound. When such a compound query is evaluated a series of relational operations are performed in order to obtain the result of the compound query. In this section, we extend the previous results from single operations to arbitrarily complex relational queries.

First, we formalize relational algebra expressions which constitute queries [20, 23]. We assume a fixed *database scheme* which is given by a finite set of relation symbols $\mathbb{r}_1, \ldots, \mathbb{r}_n$, each relation symbol $\mathbb{r}_i$ is given its relation scheme. Furthermore, we assume that all attributes appearing in the schemes of relation symbols have defined their domains. In this setting, the *relational algebra expressions* (shortly, *RA-expressions*) are defined as follows:

1. If $\mathbb{r}$ is a relation symbol on scheme R, then $\mathbb{r}$ is RA-expression on scheme R;
2. if $a \in L$, then $a_\emptyset$ is RA-expression on $\emptyset$;
3. if Q_1 and Q_2 are RA-expressions on R, then $(Q_1 \cap Q_2)$ and $(Q_1 \cup Q_2)$ are RA-expressions on R;
4. if Q_1, Q_2, and Q_3 are RA-expressions on R, then $(Q_1 \twoheadrightarrow^{Q_3} Q_2)$ is RA-expression on R;
5. if Q_1 is RA-expression on R_1 and Q_2 is RA-expression on R_2 then $(Q_1 \bowtie Q_2)$ is RA-expression on $R_1 \cup R_2$; if $R_1 \cap R_2 = \emptyset$, $c \in L$, $y_1 \in R_1$, $y_2 \in R_2$, and both attributes y_1 and y_2 have the same domain, then $(Q_1 \bowtie_{y_1 \approx y_2} Q_2)$ and $(Q_1 \bowtie_{c/y_1 \approx y_2} Q_2)$ are RA-expressions on $R_1 \cup R_2$;
6. if Q is RA-expression on R_1 and $R_2 \subseteq R_1$, then $\pi_{R_2}(Q)$ is RA-expression on R_2;
7. if Q_1 is RA-expression on R_1, Q_2 is RA-expression on $R_2 \subseteq R_1$, and Q_3 is RA-expression on $R_3 = R_1 \setminus R_2$, then $(Q_1 \div^{Q_3} Q_2)$ is RA-expression on R_3;
8. if Q is RA-expression on R, $y \in R$, and $d \in D_y$ (d is a value from the domain of y), then $\sigma_{y \approx d}(Q)$ is RA-expression on R; if $z \in R$ has the same domain as y, then $\sigma_{y \approx z}(Q)$ is RA-expression on R;
9. if Q is RA-expression on R, and f is an injective map such that $f(y)$ has the same domain as y ($y \in R$), then $\rho_f(Q)$ is RA-expression on $h(R)$.

In addition, if Q is RA-expression on R, we call R the *relation scheme of* Q.

As usual, we may evaluate RA-expressions in databases instances to get results of queries. In our setting, a *database instance* $\boldsymbol{\mathcal{D}}$ consists of RDTs which interpret the relation symbols and defines similarities on domains. In a more detail, for each relation variable $\mathbb{r}_i$ from the database scheme, a database instance $\boldsymbol{\mathcal{D}}$ defines its interpretation denoted $\mathbb{r}_i^{\boldsymbol{\mathcal{D}}}$ (an RDT) so that the relation scheme of $\mathbb{r}_i$ is the same as the scheme of $\mathbb{r}_i^{\boldsymbol{\mathcal{D}}}$. Moreover, for each attribute y, $\boldsymbol{\mathcal{D}}$ defines the similarity $\approx_y^{\boldsymbol{\mathcal{D}}}$ on its domain. The notion of database instance is presented here in a simplified form but it is sufficient for the subsequent considerations.

Given an RA-expression Q on scheme R and a database instance $\boldsymbol{\mathcal{D}}$, we denote by $Q^{\boldsymbol{\mathcal{D}}}$ the value of Q in $\boldsymbol{\mathcal{D}}$ which is an RDT on scheme R defined recursively by cases (as usual). Notice that in case of the atomic RA-expressions, we have $Q^{\boldsymbol{\mathcal{D}}} = \mathbb{r}^{\boldsymbol{\mathcal{D}}}$ whenever

$\mathbb{r}$ is a relation symbol, and $Q^{\boldsymbol{\mathcal{D}}} = a_\emptyset$ whenever Q is $a_\emptyset$. If Q is $Q_1 \cap Q_2$, then $Q^{\boldsymbol{\mathcal{D}}} = (Q_1 \cap Q_2)^{\boldsymbol{\mathcal{D}}} = Q_1^{\boldsymbol{\mathcal{D}}} \cap Q_2^{\boldsymbol{\mathcal{D}}}$ and analogously for the other cases of compound RA-expressions.

Now, we may ask the following question:

> *Do similar queries yield similar results when evaluated in similar database instances?*

By a similar query, we mean a query which results from other query by modifying some of its subqueries. For instance, if a query Q_1 involves a similarity-based restriction using constant d, we may consider its modification Q_2 by substituting d' for d and preserving the rest of this query, cf. Section 6.1.4. Then, considering two database instances $\boldsymbol{\mathcal{D}}_1$ and $\boldsymbol{\mathcal{D}}_2$, we may be interested in estimating the similarity degree $E\big(Q_1^{\boldsymbol{\mathcal{D}}_1}, Q_2^{\boldsymbol{\mathcal{D}}_2}\big)$, i.e., the degree to which $Q_1^{\boldsymbol{\mathcal{D}}_1}$ (the result of Q_1 in $\boldsymbol{\mathcal{D}}_1$) and $Q_2^{\boldsymbol{\mathcal{D}}_2}$ (the result of Q_2 in $\boldsymbol{\mathcal{D}}_2$) are similar.

In order to formalize the similarity estimates, for a pair of queries Q_1 and Q_2, we define their similarity $E(Q_1, Q_2)$. Before we show details, two clarifying notes are in order. First, there are pairs of queries for which it makes no sense to consider $E(Q_1, Q_2)$. For instance, if the relation schemes of Q_1 and Q_2 are different, we cannot express $E\big(Q_1^{\boldsymbol{\mathcal{D}}_1}, Q_2^{\boldsymbol{\mathcal{D}}_2}\big)$ by (5.1) and thus there is no point in considering its estimation. Hence, $E(Q_1, Q_2)$ may not be defined. Second, $E(Q_1, Q_2)$ is not a single degree from L. Instead, we introduce $E(Q_1, Q_2)$ as a map of the form

$$E(Q_1, Q_2) \colon \mathcal{I} \times \mathcal{I} \to L, \tag{6.47}$$

where $\mathcal{I}$ is a set of all database instances of the considered database scheme. Thus, for database instances $\boldsymbol{\mathcal{D}}_1$ and $\boldsymbol{\mathcal{D}}_2$, $\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2)$ is a degree from L. Our intention is to define the degree so that it is a lower bound of the similarity of $Q_1^{\boldsymbol{\mathcal{D}}_1}$ and $Q_2^{\boldsymbol{\mathcal{D}}_2}$.

We define (6.47) by cases taking into account the structure of Q_1 and Q_2. In the following list, we use $\stackrel{\text{def}}{=}$ to denote that the left-hand size of assignment expressions with $\stackrel{\text{def}}{=}$ is defined whenever the right-hand side is defined. Following the definition of RA-expressions, we distinguish the following cases:

- If Q_1 and Q_2 are relation symbols $\mathbb{r}_1$ and $\mathbb{r}_2$ on the same relation scheme, then

$$\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} E(\mathbb{r}_1^{\boldsymbol{\mathcal{D}}_1}, \mathbb{r}_2^{\boldsymbol{\mathcal{D}}_2}). \tag{6.48}$$

- If Q_1 and Q_2 are $a_\emptyset$ and $b_\emptyset$, then

$$\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} a \leftrightarrow b. \tag{6.49}$$

- If $Q_1 = Q_2$ and $\boldsymbol{\mathcal{D}}_1 = \boldsymbol{\mathcal{D}}_2$, then

$$\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} 1. \tag{6.50}$$

- If Q_1 is $Q_{11}\,\mathrm{op}\,Q_{12}$ and Q_2 is $Q_{21}\,\mathrm{op}\,Q_{22}$ where op in both RA-expressions is either of $\cap$, $\cup$, $\bowtie$, then

$$\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} \big(E(Q_{11}, Q_{21})\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \otimes \big(E(Q_{12}, Q_{22})\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2). \tag{6.51}$$

- If Q_1 is $\mathrm{op}(Q_1')$ and Q_2 is $\mathrm{op}(Q_2')$ where op in both RA-expressions is π_R or ρ_f, then

$$\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} \big(E(Q_1',Q_2')\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2). \tag{6.52}$$

- If Q_1 is $Q_{11}\,\mathrm{op}^{Q_{13}}\,Q_{12}$ and Q_2 is $Q_{21}\,\mathrm{op}^{Q_{23}}\,Q_{22}$ where op is $\twoheadrightarrow$ or $\div$, then

$$\begin{aligned}&\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=}\\ &\big(E(Q_{11},Q_{21})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \otimes \big(E(Q_{12},Q_{22})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \otimes \big(E(Q_{13},Q_{23})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2).\end{aligned} \tag{6.53}$$

- If Q_1 is $\sigma_{y\approx d_1}(Q_1')$ and Q_2 is $\sigma_{y\approx d_2}(Q_2')$ where $d_1, d_2 \in D_y$, then

$$\begin{aligned}&\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=}\\ &\big(E(Q_1',Q_2')\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \otimes \textstyle\bigwedge_{d\in D_y}\big(d \approx_y^{\boldsymbol{\mathcal{D}}_1} d_1 \leftrightarrow d \approx_y^{\boldsymbol{\mathcal{D}}_2} d_2\big).\end{aligned} \tag{6.54}$$

- If Q_1 is $\sigma_{y\approx y'}(Q_1')$ and Q_2 is $\sigma_{y\approx y'}(Q_2')$ where y, y' are attributes with the same domain, then

$$\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} \big(E(Q_1',Q_2')\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \otimes E\big(\approx_y^{\boldsymbol{\mathcal{D}}_1}, \approx_y^{\boldsymbol{\mathcal{D}}_2}\big). \tag{6.55}$$

- If Q_1 is $Q_{11} \bowtie_{y_1\approx y_2} Q_{12}$ and Q_2 is $Q_{21} \bowtie_{y_1\approx y_2} Q_{22}$, then

$$\begin{aligned}&\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=}\\ &\big(E(Q_{11},Q_{21})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \otimes \big(E(Q_{12},Q_{22})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \otimes E\big(\approx_y^{\boldsymbol{\mathcal{D}}_1}, \approx_y^{\boldsymbol{\mathcal{D}}_2}\big).\end{aligned} \tag{6.56}$$

- If Q_1 is $Q_{11} \bowtie_{a/y_1\approx y_2} Q_{12}$ and Q_2 is $Q_{21} \bowtie_{b/y_1\approx y_2} Q_{22}$, then

$$\begin{aligned}&\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=}\\ &(a \leftrightarrow b) \otimes \big(E(Q_{11},Q_{21})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \otimes \big(E(Q_{12},Q_{22})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \otimes E\big(\approx_y^{\boldsymbol{\mathcal{D}}_1}, \approx_y^{\boldsymbol{\mathcal{D}}_2}\big).\end{aligned} \tag{6.57}$$

The following theorem shows that similarities as defined above are indeed lower bounds of similarities of query results.

Theorem 67. *Let Q_1 and Q_2 be RA-expressions such that $E(Q_1,Q_2)$ is defined. Then, for any database instances $\boldsymbol{\mathcal{D}}_1$ and $\boldsymbol{\mathcal{D}}_2$, we have*

$$\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \le E\big(Q_1^{\boldsymbol{\mathcal{D}}_1}, Q_2^{\boldsymbol{\mathcal{D}}_2}\big). \tag{6.58}$$

Proof. The assertion is a consequence of the similarity estimates from Section 6.1 and is proved by structural induction over RA-expressions Q_1 and Q_2. □

6.3 Tuple-based similarity

In this section, we show an alternative definition of similarity of RDTs, which is connected to the notion of similarity-based closure.

While the rank-based similarity (5.1) can be sufficient in many cases there are situations in which the use of (5.1) seems to be inadequate. For example, take the RDT from Section 2.3, increase the price of every room by 1 euro and keep all other data and ranks unaltered. Then according to rank-based similarity, the original data table and the new one are very different, their similarity degree will be 0 for any choice of $\mathbf{L}$. Intuitively, since the two data tables differ only by a small change in price, one would expect to have a high degree of similarity. Hence, we wish to consider the values in tuples in addition to the ranks of tuples in RDTs when assessing similarity. Naturally, $\mathcal{D}_1$ and $\mathcal{D}_2$ will likely be considered similar if they pass a test given by the following proposition:

> *For every tuple in $\mathcal{D}_1$, there exists a similar tuple in $\mathcal{D}_2$*
> *and for every tuple in $\mathcal{D}_2$, there exists a similar tuple in $\mathcal{D}_1$.*

That is, one may define

$$S^{\approx}(\mathcal{D}_1, \mathcal{D}_2) = \bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \to \bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}_2(r') \otimes r \approx_R r'\big)\big), \tag{6.59}$$

$$E^{\approx}(\mathcal{D}_1, \mathcal{D}_2) = S^{\approx}(\mathcal{D}_1, \mathcal{D}_2) \wedge S^{\approx}(\mathcal{D}_2, \mathcal{D}_1), \tag{6.60}$$

where the similarity degree $r \approx_R r'$ of tuples r and r' is defined by

$$r \approx_R r' = \bigwedge_{y \in R} r(y) \approx_y r'(y). \tag{6.61}$$

Note that (6.61) is a particular definition of a degree to which tuples r and r' have similar values. Namely, $r \approx_R r'$ is a degree to which "for each attribute $y \in R$, $r(y)$ is similar to $r'(y)$". Observe that $\approx_R$ defined by (6.61) is reflexive and symmetric since each $\approx_y$ is reflexive and symmetric. Moreover, if all $\approx_y$ are $\otimes$-transitive ($\mathbf{L}$-equivalences), then $\approx_R$ is $\otimes$-transitive ($\mathbf{L}$-equivalence) as well.

Remark 24. *(i) Note that* (6.59) *is used in [9, Section 4.2] to assess similarity between two fuzzy sets in a universe equipped with a similarity relation. (ii) As in Remark 19, if* $\mathbf{L}$ *is the two-valued Boolean algebra and each $\approx_y$ is an identity, then $E^{\approx}(\mathcal{D}_1, \mathcal{D}_2) = 1$ iff $\mathcal{D}_1$ and $\mathcal{D}_2$ are identical (in the usual sense). (iii) Obviously, $S(\mathcal{D}_1, \mathcal{D}_2) \leq S^{\approx}(\mathcal{D}_1, \mathcal{D}_2)$ due to the isotony of $\to$ in the second argument* (2.5) *and the reflexivity of $\approx_R$. Therefore,* (6.59) *yields an estimate which is at least as high as* (5.2)*; analogously for E and $E^{\approx}$.*

6.3.1 Similarity-based semijoins and closures

The similarity of RDTs based on (6.59) can be expressed using (5.2) and a derived relational operation. In order to show this relationship, we digress and introduce further relational operations which involve similarity of tuple values [23].

For any RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$ on $R_1 \cup R_2$ and $R_2 \cup R_3$ such that R_1, R_2, and R_3 are pairwise disjoint, a (*natural*) *similarity-based semijoin* $\mathcal{D}_1 \ltimes^{\approx} \mathcal{D}_2$ of $\mathcal{D}_1$ and $\mathcal{D}_2$ is defined by

$$\big(\mathcal{D}_1 \ltimes^{\approx} \mathcal{D}_2\big)(r_1 r_2) = \mathcal{D}_1(r_1 r_2) \otimes \bigvee_{r_2' \in \mathrm{Tupl}(R_2)} \bigvee_{r_3 \in \mathrm{Tupl}(R_3)} \big(\mathcal{D}_2(r_2' r_3) \otimes r_2' \approx_{R_2} r_2\big), \quad (6.62)$$

for all $r_1 \in \mathrm{Tupl}(R_1)$ and $r_2 \in \mathrm{Tupl}(R_2)$. If $\mathcal{D}_1$ and $\mathcal{D}_2$ are viewed as results of queries Q_1 and Q_2, then (6.62) is a degree to which $r_1 r_2$ from $\mathcal{D}_1$ approximately matches a tuple from $\mathcal{D}_2$ (namely, $r_1 r_2$ matches Q_1 and r_2 is similar to r_2' for which $r_2' r_3$ matches Q_2). Clearly, (6.62) is a similarity-based counterpart of the ordinary semijoin.

It can be shown that similarity-based semijoins are indeed derived relational operations [23]. Now we will consider a particular case of similarity-based semijoins. Namely, we let $R_1 = R_3 = \emptyset$ and consider $\mathcal{D}_1$ to be nonranked such that $\mathcal{D}_2 \subseteq \mathcal{D}_1$. In this particular case, we denote $\mathcal{D}_1 \ltimes^{\approx} \mathcal{D}_2$ by $\mathrm{C}^{\approx}_{\mathcal{D}_1}(\mathcal{D}_2)$ and call it a *similarity closure of* $\mathcal{D}_2$ (*with respect to* $\mathcal{D}_1$). Thus,

$$(\mathrm{C}^{\approx}_{\mathcal{D}_1}(\mathcal{D}_2))(r) = \mathcal{D}_1(r) \otimes \bigvee_{r' \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_2(r') \otimes r' \approx_R r\big) \quad (6.63)$$

for each $r \in \mathrm{Tupl}(R_2)$. Furthermore, since $\mathcal{D}_1$ is nonranked, we may write

$$(\mathrm{C}^{\approx}_{\mathcal{D}_1}(\mathcal{D}_2))(r) = \bigvee_{r' \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_2(r') \otimes r' \approx_R r\big), \quad (6.64)$$

whenever $r \in \mathcal{D}_1$ (and $= 0$ otherwise).

Taking into account (6.63), $\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)$ can be seen as a result of query: "Show all tuples which are in $\mathcal{D}_1$ and, in addition, include all tuples which are from $\mathcal{D}$ and are similar to those in $\mathcal{D}_1$." Thus, $\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)$ extends $\mathcal{D}_1$ by all tuples from $\mathcal{D}$ similar to tuples from $\mathcal{D}_1$. This can be useful in many cases, especially in the query-by-example paradigm. For example $\mathcal{D}_1$ can store information about ideal candidates for a particular job position with ranks indicating the degree to which the ideal candidates satisfy our requirements. Suppose the data table $\mathcal{D}$ contains information about real job applicants. Then $\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)$ will give us a collection of job applicants satisfying our requirements given by examples in $\mathcal{D}_1$ together with the degree of satisfaction (rank). More precisely $\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)$ will contain (among the ideal candidates) a collection of job applicants for which there exists a similar ideal candidate.

Similarity-based closures and semijoins may be considered as examples of nontrivial relational operations which do not appear in the classical relational model. More precisely, they do appear in the Codd model but only in a trivial form—$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)$ equals $\mathcal{D}_1$ and the similarity-based semijoin coincides with the ordinary semijoin. Also note that (6.63) can be seen as a domain independent variant of a concept of a similarity closure which appears in fuzzy relational systems [9, 10, 73].

Using similarity-closures, it is now apparent that $S^{\approx}$ defined by (6.59) can be restated as

$$S^{\approx}(\mathcal{D}_1, \mathcal{D}_2) = \bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \to \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)(r)\big)$$

$$= S\big(\mathcal{D}_1, \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)\big), \tag{6.65}$$

where $\mathcal{D}$ is nonranked such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$. Clearly, the value of (6.65) does not depend on the choice of a nonranked $\mathcal{D}$ satisfying $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$.

If $\approx_R$ is $\otimes$-transitive, then $\mathrm{C}^{\approx}_{\mathcal{D}}$ forms an **L**-closure operator [9, 8]:

Lemma 68. *Let $\mathcal{D}$ be a nonranked table on R and let $\approx_R$ be $\otimes$-transitive. Then, $\mathrm{C}^{\approx}_{\mathcal{D}}$ is an* **L***-closure operator, i.e., it satisfies*

$$\mathcal{D}_1 \subseteq \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1), \tag{6.66}$$

$$S(\mathcal{D}_1, \mathcal{D}_2) \leq S(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1), \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)), \tag{6.67}$$

$$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)) = \mathrm{C}^{\approx}_{\mathcal{D}}(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)), \tag{6.68}$$

for all $\mathcal{D}_1$ and $\mathcal{D}_2$ on R such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$.

Proof. The assertion is proved using the same arguments as in [8]. □

Remark 25. *Note that if $\approx_R$ is not $\otimes$-transitive, then $\mathrm{C}^{\approx}_{\mathcal{D}}$ still satisfies* (6.66) *and* (6.67) *but it is not idempotent in general. Indeed, consider $\mathcal{D}$ on $\{y\}$ and $\approx_y$ from Remark 22 and a nonranked $\mathcal{D}' \supseteq \mathcal{D}$ which consists c, d, d' as y-values of some of its tuples with nonzero ranks. In addition, take r, r' such that $r(y) = d$ and $r'(y) = d'$. By definition, $(\mathrm{C}^{\approx}_{\mathcal{D}'}(\mathcal{D}))(r') = 0$ because $c \approx_y d' = 0$. On the other hand, $(\mathrm{C}^{\approx}_{\mathcal{D}'}(\mathcal{D}))(r) = 0.8$ because $c \approx_y d = 0.8$ and thus using $d \approx_y d' = 0.9$ we get $(\mathrm{C}^{\approx}_{\mathcal{D}'}(\mathrm{C}^{\approx}_{\mathcal{D}'}(\mathcal{D})))(r') = 0.8 \otimes 0.9 = 0.7 > (\mathrm{C}^{\approx}_{\mathcal{D}'}(\mathcal{D}))(r')$. As a consequence,* (6.68) *is not satisfied.*

Moreover, as a consequence of Lemma 68, we get

$$\begin{aligned} S^{\approx}(\mathcal{D}_1, \mathcal{D}_2) &= S(\mathcal{D}_1, \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)) \\ &= S(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1), \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)) \\ &= S^{\approx}(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1), \mathcal{D}_2) \end{aligned} \tag{6.69}$$

and analogously for E provided that $\approx_R$ is $\otimes$-transitive. Indeed, the "$\leq$"-part of (6.69) follows from (6.67) and (6.68); the "$\geq$"-part follows from (6.66) and antitony of $\to$ in the first argument.

Based on our observations, we may view $S^{\approx}$ and $E^{\approx}$ as being defined using (5.2), (5.1), and similarity-based closures of RDTs. The following theorem shows further properties of $\mathrm{C}^{\approx}_{\mathcal{D}}$ with respect to other relational operations.

Theorem 69. *For any RDTs $\mathcal{D}_1, \mathcal{D}_2$ and nonranked $\mathcal{D}$ on R, such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$:*

$$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \cup \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2) = \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1 \cup \mathcal{D}_2), \tag{6.70}$$

$$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1 \cap \mathcal{D}_2) \subseteq \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \cap \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2), \tag{6.71}$$

If $\mathcal{D}_1$ and $\mathcal{D}_2$ are RDTs on disjoint schemes R_1 and R_2, respectively, $R \subseteq R_1$, then

$$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \bowtie \mathrm{C}^{\approx}_{\mathcal{D}'}(\mathcal{D}_2) \subseteq \mathrm{C}^{\approx}_{\mathcal{D} \bowtie \mathcal{D}'}(\mathcal{D}_1 \bowtie \mathcal{D}_2), \tag{6.72}$$

$$\pi_R(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)) \subseteq \mathrm{C}^{\approx}_{\pi_R(\mathcal{D})}(\pi_R(\mathcal{D}_1)), \tag{6.73}$$

for any nonranked $\mathcal{D}$ and $\mathcal{D}'$ on R_1 and R_2, respectively, such that $\mathcal{D}_1 \subseteq \mathcal{D}$ and $\mathcal{D}_2 \subseteq \mathcal{D}'$.

Proof. The "$\subseteq$"-part of (6.70) is a consequence of isotony of $\mathrm{C}^{\approx}_{\mathcal{D}}$. In order to prove the "$\supseteq$"-part, observe that

$$
\begin{aligned}
&(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1 \cup \mathcal{D}_2))(r) = \\
&\mathcal{D}(r) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big((\mathcal{D}_1(r') \vee \mathcal{D}_2(r')) \otimes r' \approx_R r\big) = \\
&\mathcal{D}(r) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big((\mathcal{D}_1(r') \otimes r' \approx_R r) \vee (\mathcal{D}_2(r') \otimes r' \approx_R r)\big) \leq \\
&\mathcal{D}(r) \otimes \big(\textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r') \otimes r' \approx_R r\big) \vee \bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}_2(r') \otimes r' \approx_R r\big)\big) = \\
&\big(\mathcal{D}(r) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r') \otimes r' \approx_R r\big)\big) \vee \big(\mathcal{D}(r) \otimes \bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}_2(r') \otimes r' \approx_R r\big)\big) = \\
&(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1))(r) \vee (\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2))(r) = (\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \cup \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2))(r)
\end{aligned}
$$

for any $r \in \mathrm{Tupl}(R)$, showing (6.70);
(6.71): Consequence of isotony of $\mathrm{C}^{\approx}_{\mathcal{D}}$;
(6.72): Using (2.12) we get

$$
\begin{aligned}
&(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \bowtie \mathrm{C}^{\approx}_{\mathcal{D}'}(\mathcal{D}_2))(r_1 r_2) = \\
&\big(\mathcal{D}(r_1) \otimes \textstyle\bigvee_{r_1' \in \mathrm{Tupl}(R_1)} \big(\mathcal{D}_1(r_1') \otimes r_1' \approx_{R_1} r_1\big)\big) \otimes \big(\mathcal{D}'(r_2) \otimes \bigvee_{r_2' \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_2(r_2') \otimes r_2' \approx_{R_2} r_2\big)\big) = \\
&\mathcal{D}(r_1) \otimes \mathcal{D}'(r_2) \otimes \textstyle\bigvee_{r_1' \in \mathrm{Tupl}(R_1)} \bigvee_{r_2' \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_1(r_1') \otimes \mathcal{D}_2(r_2') \otimes r_1' \approx_{R_1} r_1 \otimes r_2' \approx_{R_2} r_2\big) \leq \\
&\mathcal{D}(r_1) \otimes \mathcal{D}'(r_2) \otimes \textstyle\bigvee_{r_1' \in \mathrm{Tupl}(R_1)} \bigvee_{r_2' \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_1(r_1') \otimes \mathcal{D}_2(r_2') \otimes (r_1' \approx_{R_1} r_1 \wedge r_2' \approx_{R_2} r_2)\big) = \\
&(\mathcal{D} \bowtie \mathcal{D}')(r_1 r_2) \otimes \textstyle\bigvee_{r_1' \in \mathrm{Tupl}(R_1)} \bigvee_{r_2' \in \mathrm{Tupl}(R_2)} \big((\mathcal{D}_1 \bowtie \mathcal{D}_2)(r_1' r_2') \otimes (r_1' \approx_{R_1} r_1 \wedge r_2' \approx_{R_2} r_2)\big) = \\
&(\mathcal{D} \bowtie \mathcal{D}')(r_1 r_2) \otimes \textstyle\bigvee_{r_1' r_2' \in \mathrm{Tupl}(R_1 \cup R_2)} \big((\mathcal{D}_1 \bowtie \mathcal{D}_2)(r_1' r_2') \otimes r_1' r_2' \approx_{R_1 \cup R_2} r_1 r_2\big) = \\
&\big(\mathrm{C}^{\approx}_{\mathcal{D} \bowtie \mathcal{D}'}(\mathcal{D}_1 \bowtie \mathcal{D}_2)\big)(r_1 r_2).
\end{aligned}
$$

(6.73): Taking into account (2.53), it suffices to check that

$$
\begin{aligned}
&(\pi_R(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)))(r) = \\
&\textstyle\bigvee_{r_2 \in \mathrm{Tupl}(R_1 \setminus R)} (\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1))(r r_2) \leq \big(\mathrm{C}^{\approx}_{\pi_R(\mathcal{D})}(\pi_R(\mathcal{D}_1))\big)(r)
\end{aligned}
$$

for all $r \in \mathrm{Tupl}(R)$. Thus, is suffices to show $(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1))(r r_2) \leq (\mathrm{C}^{\approx}_{\pi_R(\mathcal{D})}(\pi_R(\mathcal{D}_1)))(r)$ for all $r \in \mathrm{Tupl}(R)$ and $r_2 \in \mathrm{Tupl}(R_1 \setminus R)$, which is indeed true:

$$
\begin{aligned}
&(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1))(r r_2) = \\
&\mathcal{D}(r r_2) \otimes \textstyle\bigvee_{r' r_2' \in \mathrm{Tupl}(R_1)} \big(\mathcal{D}_1(r' r_2') \otimes r' r_2' \approx_{R_1} r r_2\big) \leq \\
&\mathcal{D}(r r_2) \otimes \textstyle\bigvee_{r' r_2' \in \mathrm{Tupl}(R_1)} \big(\mathcal{D}_1(r' r_2') \otimes r' \approx_R r\big) = \\
&\mathcal{D}(r r_2) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big(\bigvee_{r_2' \in \mathrm{Tupl}(R_1 \setminus R)} \mathcal{D}_1(r' r_2') \otimes r' \approx_R r\big) = \\
&\mathcal{D}(r r_2) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big((\pi_R(\mathcal{D}))(r') \otimes r' \approx_R r\big) \leq \\
&\textstyle\bigvee_{r_2 \in \mathrm{Tupl}(R_1 \setminus R)} \mathcal{D}(r r_2) \otimes \bigvee_{r' \in \mathrm{Tupl}(R)} \big((\pi_R(\mathcal{D}))(r') \otimes r' \approx_R r\big) = \\
&(\pi_R(\mathcal{D}))(r) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big((\pi_R(\mathcal{D}))(r') \otimes r' \approx_R r\big) = \\
&\big(\mathrm{C}^{\approx}_{\pi_R(\mathcal{D})}(\pi_R(\mathcal{D}_1))\big)(r).
\end{aligned}
$$

□

6.3.2 Tuple-based similarity estimates

As in case of the rank-based similarity introduced in Section 5.1, we may investigate inequalities which provide tuple-based similarity estimates of query results based on input data. Unlike the rank-based approach, the tuple-based approach has some limitations. In this section, we provide an if and only if criterion for general relational operations which preserve tuple-based similarity. The criterion is based on similarity closures introduced in the previous section.

In the section, we make the following assumptions. We consider relation schemes $R_1, \ldots, R_n, R$ and a map f which maps any RDTs $\mathcal{D}_1, \ldots, \mathcal{D}_n$ on $R_1, \ldots, R_n$ to an RDT $f(\mathcal{D}_1, \ldots, \mathcal{D}_n)$ on R (called the result of f). The map f represents a general n-ary relational operation for which we investigate the issues related to preservation of tuple-based similarity.

Furthermore, let $\odot$ be a binary operation on L with 1 being its neutral element. The operation f is called *S-compatible with respect to $\odot$* if for some $0 \leq j \leq n$, we have

$$\bigodot_{i=1}^{j} S(\mathcal{D}_i, \mathcal{D}'_i) \odot \bigodot_{i=j+1}^{n} S(\mathcal{D}'_i, \mathcal{D}_i) \leq S\big(f(\mathcal{D}_1, \ldots, \mathcal{D}_n), f(\mathcal{D}'_1, \ldots, \mathcal{D}'_n)\big), \tag{6.74}$$

for all $\mathcal{D}_i, \mathcal{D}'_i$ on R_i $(i = 1, \ldots, n)$. Analogously, f is called *$S^{\approx}$-compatible with respect to $\odot$* if (6.74) holds for S replaced by $S^{\approx}$. Furthermore, f is called *E-compatible* and *$E^{\approx}$-compatible* if (6.74) holds for S replaced by E and $E^{\approx}$, respectively.

In a sense, (6.74) generalizes the condition from Lemma 49. From the point of view of tuple-based similarity, it is interesting to investigate the relationship between S-compatibility (E-compatibility) and $S^{\approx}$-compatibility ($E^{\approx}$-compatibility). The following theorem gives an if-and-only-if criterion for a general n-ary operation on ranked data tables to be compatible with tuple-based inclusion $S^{\approx}$.

Theorem 70. *Let f be S-compatible with respect to $\odot$. Then, the following statements are equivalent:*

(i) For any $\mathcal{D}_1, \ldots, \mathcal{D}_n$ and nonranked $\mathcal{D}'_1, \ldots, \mathcal{D}'_n$ such that $\mathcal{D}_1 \subseteq \mathcal{D}'_1, \ldots, \mathcal{D}_n \subseteq \mathcal{D}'_n$ there is a nonranked $\mathcal{D}$ such that $f(\mathcal{D}_1, \ldots, \mathcal{D}_n) \subseteq \mathcal{D}$ and

$$f\big(\mathrm{C}^{\approx}_{\mathcal{D}'_1}(\mathcal{D}_1), \ldots, \mathrm{C}^{\approx}_{\mathcal{D}'_n}(\mathcal{D}_n)\big) \subseteq \mathrm{C}^{\approx}_{\mathcal{D}}(f(\mathcal{D}_1, \ldots, \mathcal{D}_n));$$

(ii) f is $S^{\approx}$-compatible with respect to $\odot$.

Proof. For simplicity of presentation of the proof, we only consider (6.74) for $j = n$ (the general case can be proved using the same arguments). In order to prove the only-if part, observe that the S-compatibility with respect to $\odot$ yields

$$\begin{aligned}
&S^{\approx}(\mathcal{D}_1, \mathcal{D}'_1) \odot \cdots \odot S^{\approx}(\mathcal{D}_n, \mathcal{D}'_n) = \\
&S\big(\mathcal{D}_1, \mathrm{C}^{\approx}_{\mathcal{D}''_1}(\mathcal{D}'_1)\big) \odot \cdots \odot S\big(\mathcal{D}_n, \mathrm{C}^{\approx}_{\mathcal{D}''_n}(\mathcal{D}'_n)\big) \leq \\
&S\big(f(\mathcal{D}_1, \ldots, \mathcal{D}_n), f(\mathrm{C}^{\approx}_{\mathcal{D}''_1}(\mathcal{D}'_1), \ldots, \mathrm{C}^{\approx}_{\mathcal{D}''_n}(\mathcal{D}'_n))\big).
\end{aligned}$$

Moreover, from (i) and the isotony of $\to$ in the second argument, it follows that

$$\begin{aligned}
&S\big(f(\mathcal{D}_1,\ldots,\mathcal{D}_n), f(\mathrm{C}^{\approx}_{\mathcal{D}''_1}(\mathcal{D}'_1),\ldots,\mathrm{C}^{\approx}_{\mathcal{D}''_n}(\mathcal{D}'_n))\big) \leq \\
&S\big(f(\mathcal{D}_1,\ldots,\mathcal{D}_n), \mathrm{C}^{\approx}_{\mathcal{D}}(f(\mathcal{D}'_1,\ldots,\mathcal{D}'_n))\big) = \\
&S^{\approx}\big(f(\mathcal{D}_1,\ldots,\mathcal{D}_n), f(\mathcal{D}'_1,\ldots,\mathcal{D}'_n)\big).
\end{aligned}$$

Hence, (ii) follows from (i).

Conversely, let (ii) holds and take $\mathcal{D}_1,\ldots,\mathcal{D}_n$ and nonranked $\mathcal{D}'_1,\ldots,\mathcal{D}'_n$ where $\mathcal{D}_1 \subseteq \mathcal{D}'_1,\ldots,\mathcal{D}_n \subseteq \mathcal{D}'_n$. Applying the $S^{\approx}$ compatibility with respect to $\odot$,

$$\begin{aligned}
&S^{\approx}\big(\mathrm{C}^{\approx}_{\mathcal{D}'_1}(\mathcal{D}_1),\mathcal{D}_1\big) \odot \cdots \odot S^{\approx}\big(\mathrm{C}^{\approx}_{\mathcal{D}'_n}(\mathcal{D}_n),\mathcal{D}_n\big) \leq \\
&S^{\approx}\big(f\big(\mathrm{C}^{\approx}_{\mathcal{D}'_1}(\mathcal{D}_1),\ldots,\mathrm{C}^{\approx}_{\mathcal{D}'_n}(\mathcal{D}_n)\big), f(\mathcal{D}_1,\ldots,\mathcal{D}_n)\big) = \\
&S\big(f\big(\mathrm{C}^{\approx}_{\mathcal{D}'_1}(\mathcal{D}_1),\ldots,\mathrm{C}^{\approx}_{\mathcal{D}'_n}(\mathcal{D}_n)\big), \mathrm{C}^{\approx}_{\mathcal{D}}(f(\mathcal{D}_1,\ldots,\mathcal{D}_n))\big)
\end{aligned}$$

for some nonranked $\mathcal{D}$ such that $f(\mathcal{D}_1,\ldots,\mathcal{D}_n) \subseteq \mathcal{D}$. Moreover, for each $i = 1,\ldots,n$, we have

$$S^{\approx}\big(\mathrm{C}^{\approx}_{\mathcal{D}'_i}(\mathcal{D}_i),\mathcal{D}_i\big) = S^{\approx}\big(\mathrm{C}^{\approx}_{\mathcal{D}'_i}(\mathcal{D}_i),\mathrm{C}^{\approx}_{\mathcal{D}'_i}(\mathcal{D}_i)\big) = 1.$$

Since, $\odot$ is neutral with respect to 1, the previous inequality yields

$$S\big(f\big(\mathrm{C}^{\approx}_{\mathcal{D}'_1}(\mathcal{D}_1),\ldots,\mathrm{C}^{\approx}_{\mathcal{D}'_n}(\mathcal{D}_n)\big), \mathrm{C}^{\approx}_{\mathcal{D}}(f(\mathcal{D}_1,\ldots,\mathcal{D}_n))\big) = 1$$

which proves (i). □

Theorem 70 enables us to simplify proofs for $S^{\approx}_{\mathcal{D}}$-compatibility of relational operations. In order to prove that operation f is $S^{\approx}$-compatible, it is sufficient to show that f is S-compatible together with (i) of Theorem 70. The following corollary is a consequence of Theorem 70, Theorem 69 and the S-compatibility of operations with respect to $\otimes$ or $\wedge$ proved in Section 6.1.

Corollary 71. *The following inequalities*

$$\begin{aligned}
&S^{\approx}(\mathcal{D}_1,\mathcal{D}_2) \leq S^{\approx}(\pi_R(\mathcal{D}_1),\pi_R(\mathcal{D}_2)), \\
&S^{\approx}(\mathcal{D}_1,\mathcal{D}'_1) \wedge S^{\approx}(\mathcal{D}_2,\mathcal{D}'_2) \leq S^{\approx}(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2), \\
&S^{\approx}(\mathcal{D}_1,\mathcal{D}_2) \otimes S^{\approx}(\mathcal{D}_3,\mathcal{D}_4) \leq S^{\approx}(\mathcal{D}_1 \bowtie \mathcal{D}_3, \mathcal{D}_2 \bowtie \mathcal{D}_4),
\end{aligned}$$

hold for any RDTs provided that the relations schemes of $\mathcal{D}_1$ and $\mathcal{D}_3$ ($\mathcal{D}_2$ and $\mathcal{D}_4$) are disjoint. Moreover, the same inequalities hold if $S^{\approx}$ is replaced by $E^{\approx}$. □

It can be shown by means of simple counterexamples that relational operations from previous sections excluding those listed in Corollary 71 are not $S^{\approx}$-compatible with respect to $\otimes$. We will now show such counterexamples for some relational operations. In all of the following examples we utilize the residuated lattice with $L = [0.1, 0.2, \ldots, 1]$ given by the Łukasiewicz operations together with * being identity. We consider domains $D_y = D_z = \{0.1, 0.2, \ldots, 1\}$, with similarities given by biresiduum, i.e. $a_1 \approx_y a_2 = a_1 \leftrightarrow a_2$ for any $y \in R, a_1, a_2 \in D_y$.

Example 6. *Let us start with selection, which was proven to be S-compatible by Theorem 56. According to Theorem 70, for proving $S^{\approx}$-compatibility of selection we need to show that $\sigma_{y\approx d}(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)) \subseteq \mathrm{C}^{\approx}_{\mathcal{D}}(\sigma_{y\approx d}(\mathcal{D}_1))$. The following choice of RDTs falsifies this claim and therefore selection is not $S^{\approx}$-compatible. Consider domain D_y and $d = 0.3 \in D_y$, $\mathcal{D}$, and $\mathcal{D}_1$ given as below.*

$\mathcal{D}$

	y
1	0.1
1	0.3

$\mathcal{D}_1$

	y
0.5	0.1

$\sigma_{y\approx 0.3}(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1))$

	y
0.3	0.1
0.3	0.3

$\mathrm{C}^{\approx}_{\mathcal{D}}(\sigma_{y\approx 0.3}(\mathcal{D}_1))$

	y
0.3	0.1
0.1	0.3

Example 7. *Similarly, ternary residuum $\twoheadrightarrow$ is $S^{\approx}$-compatible iff $\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \twoheadrightarrow^{\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_3)} \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2) \subseteq \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2)$ for any $\mathcal{D}_1$, $\mathcal{D}'_1$, $\mathcal{D}_2$, $\mathcal{D}'_2$, $\mathcal{D}_3$, and $\mathcal{D}'_3$ on relation scheme R. The following choice of RDTs demonstrates that the claim doesn't hold in general.*

$\mathcal{D} = \mathcal{D}_3$

	y	z
1	0.8	0.6
1	0.8	0.5
1	0.1	0.6
1	0.1	0.5

$\mathcal{D}_1$

	y	z
0.7	0.8	0.6
0.6	0.1	0.6
0.4	0.1	0.5

$\mathcal{D}_2$

	y	z
0.9	0.1	0.6
0.2	0.8	0.6

$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \twoheadrightarrow^{\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_3)} \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)$

	y	z
1.0	0.1	0.6
1.0	0.1	0.5
0.5	0.8	0.6
0.6	0.8	0.5

$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2)$

	y	z
1.0	0.1	0.6
1.0	0.8	0.5
0.9	0.8	0.6
0.9	0.1	0.5

In the last example we will show that even $\otimes$ is not $S^{\approx}$-compatible.

Example 8. *Neither $\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1 \otimes \mathcal{D}_2) \subseteq \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \otimes \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)$ or $\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1 \otimes \mathcal{D}_2) \supseteq \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \otimes \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)$ hold in general (for any $\mathbf{L}$ and arbitrary similarities on domains).*

$\mathcal{D}$

	y	z
1	0.4	0.2
1	0.8	0.2
1	0.4	0.7
1	0.8	0.7

$\mathcal{D}_1$

	y	z
0.9	0.4	0.2
0.5	0.8	0.7

$\mathcal{D}_2$

	y	z
1.0	0.8	0.7
0.3	0.4	0.7

$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \otimes \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)$

	y	z
0.5	0.8	0.7
0.4	0.4	0.2

$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1 \otimes \mathcal{D}_2)$

	y	z
0.5	0.8	0.7
0.1	0.4	0.7

Analogous observation can be made for division or intersection.

We have seen that for many relational operations we cannot provide inclusion and similarity estimates (6.74). On the other hand, the relational operations may be extended to satisfy the $S^{\approx}$-compatibility if all $\approx_R$ are $\otimes$-transitive. For instance, consider (2.55) and observe that using (6.24) and (6.69), we get

$$\begin{aligned} S^{\approx}(\mathcal{D}_1, \mathcal{D}_2) &= S(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1), \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)) \\ &\leq S(\sigma_{y\approx d}(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)), \sigma_{y\approx d}(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2))) \\ &\leq S^{\approx}(\sigma_{y\approx d}(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)), \sigma_{y\approx d}(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2))). \end{aligned}$$

The result of $\sigma_{y\approx d}(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1))$ can be seen as a new relational operation—an extended restriction which is *compatible with* $S^{\approx}$ and select results not only from tuples in $\mathcal{D}_1$ but also from tuples in $\mathcal{D}$ which are similar to those in $\mathcal{D}_1$. In a similar fashion, one may proceed with the other operations and derive new $S^{\approx}$-compatible variants of the relational operations.

6.3.3 Unifying approach to similarity of RDTs

It was shown in [12] that both (5.2) and (6.65) have a common generalization using truth-stressing hedge. Let * be truth-stressing hedge on $\mathbf{L}$. For RDTs $\mathcal{D}_1, \mathcal{D}_2$ on R and nonranked $\mathcal{D}$ on R such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$ we define a degree $S^{\approx}_*(\mathcal{D}_1, \mathcal{D}_2)$ of inclusion of $\mathcal{D}_1$ in $\mathcal{D}_2$ (with respect to *) and a degree of similarity $E^{\approx}_*(\mathcal{D}_1, \mathcal{D}_2)$ with respect to * as

$$S^{\approx}_*(\mathcal{D}_1, \mathcal{D}_2) = \textstyle\bigwedge_{r\in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \to \mathrm{C}^{\approx^*}_{\mathcal{D}}(\mathcal{D}_2)\big), \tag{6.75}$$

$$E^{\approx}_*(\mathcal{D}_1, \mathcal{D}_2) = S^{\approx}_*(\mathcal{D}_1, \mathcal{D}_2) \wedge S^{\approx}_*(\mathcal{D}_2, \mathcal{D}_1), \tag{6.76}$$

where $\mathrm{C}^{\approx^*}_{\mathcal{D}}(\mathcal{D}_2)$ is a similarity-based closure of $\mathcal{D}_2$ with respect to $\mathcal{D}$ and hedge $*$ and is defined as

$$(\mathrm{C}^{\approx^*}_{\mathcal{D}}(\mathcal{D}_2))(r) = \mathcal{D}(r) \otimes \textstyle\bigvee_{r'\in \mathrm{Tupl}(R)} \big(\mathcal{D}_2(r') \otimes (r' \approx r)^*\big). \tag{6.77}$$

If there is no confusion, we will denote $\mathrm{C}^{\approx^*}_{\mathcal{D}}(\mathcal{D}_2)$ by $\mathrm{C}^{*}_{\mathcal{D}}(\mathcal{D}_2)$.

Now, observe that for $*$ being the identity, (6.75) coincides with (6.59). Furthermore if $\approx_R$ is separating (i.e., $r_1 \approx r_2 = 1$ iff r_1 is identical to r_2) and $*$ is the globalization, (6.75) coincides with (5.2). Thus, both rank-based similarity (5.1) and tuple-based similarity (6.60) are particular instances of (6.76). Therefore, the hedge in (6.75) serves as a parameter, and determines how much emphasis we put on the fact that two tuples are similar. In case of globalization, we put full emphasis, i.e. the tuples are required to be equal to degree 1.

The following lemma shows that $S^{\approx}_*$ and consequently $E^{\approx}_*$ have properties that are considered natural for (degrees of) inclusion and similarity:

Lemma 72. *If $\approx$ satisfies $(r \approx s)^* \otimes (s \approx t)^* \leq (r \approx t)^*$ with respect to * then*

(i) $S^{\approx}_*$ is a reflexive and transitive $\mathbf{L}$*-relation, i.e. an* $\mathbf{L}$*-quasiorder.*

(*ii*) $E_*^{\approx}$ *is an* **L***-equivalence.*

Proof. The assertion follows from results in [9, Section 4.2]. □

The condition $(r \approx s)^* \otimes (s \approx t)^* \leq (r \approx t)^*$ is satisfied (among other cases) in the following situations:

i) $*$ is globalization and $\approx$ is separating. If $(r \approx s)^* \otimes (s \approx t)^*$ is nonzero, then $r \approx s = 1$ and $s \approx t = 1$. Separability implies $r = s = t$, i.e. $(r \approx t)^* = 1^* = 1$.

ii) $\approx$ is transitive. In this case, since $a^* \otimes b^* \leq (a \otimes b)^*$, transitivity of $\approx$ and monotony of * yield $(r \approx s)^* \otimes (s \approx t)^* \leq ((r \approx s) \otimes (s \approx t))^* \leq (r \approx t)^*$.

By considering two different hedges $*_1, *_2$ on **L** we obtain for any RDTs two different subsethood degrees (and two similarity degrees), one using $*_1$ and one using $*_2$. We will denote such degree $S_{*_1}^{\approx}$ and $S_{*_2}^{\approx}$. In the rest of this section, we will investigate the role of hedge and the relationship between $S_{*_1}^{\approx}$ and $S_{*_2}^{\approx}$. First of all, note that if $*_1$ is stronger than $*_2$, see Section 2, then $a^{*_1} \leq a^{*_2}$ for any $a \in L$. As an immediate consequence we obtain the following lemma, which states that stronger hedge yields smaller closure and, as a consequence, smaller subsethood and similarity degrees.

Lemma 73. *Let $*_1, *_2$ be two different hedges on* **L** *such that $\mathit{fix}(*_1) \subseteq \mathit{fix}(*_2)$. Then for any RDTs $\mathcal{D}_1$, $\mathcal{D}_2$ on R and any nonranked $\mathcal{D}$ on R such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$:*

$$\mathrm{C}_{\mathcal{D}}^{*_1}(\mathcal{D}_1) \subseteq \mathrm{C}_{\mathcal{D}}^{*_2}(\mathcal{D}_1), \tag{6.78}$$

$$S_{*_1}^{\approx}(\mathcal{D}_1, \mathcal{D}_2) \leq S_{*_2}^{\approx}(\mathcal{D}_1, \mathcal{D}_2), \tag{6.79}$$

$$E_{*_1}^{\approx}(\mathcal{D}_1, \mathcal{D}_2) \leq E_{*_2}^{\approx}(\mathcal{D}_1, \mathcal{D}_2). \tag{6.80}$$

Since globalization is the strongest hedge with $\{0, 1\}$ being the only fixed points and identity is the weakest hedge on **L**, we have that for any hedge $*$:

$$S(\mathcal{D}_1, \mathcal{D}_2) \leq S_*^{\approx}(\mathcal{D}_1, \mathcal{D}_2) \leq S^{\approx}(\mathcal{D}_1, \mathcal{D}_2). \tag{6.81}$$

The following lemma shows that similar hedges yield similar closures.

Lemma 74. *Let $*_1, *_2$ be two hedges on* **L**. *Then for any RDT $\mathcal{D}'$ and any nonranked RDT $\mathcal{D}$ on R such that $\mathcal{D}' \subseteq \mathcal{D}$ we have:*

$$(*_1 \preceq *_2) \leq S(\mathrm{C}_{\mathcal{D}}^{*_1}(\mathcal{D}'), \mathrm{C}_{\mathcal{D}}^{*_2}(\mathcal{D}')), \tag{6.82}$$

$$(*_1 \approx *_2) \leq E(\mathrm{C}_{\mathcal{D}}^{*_1}(\mathcal{D}'), \mathrm{C}_{\mathcal{D}}^{*_2}(\mathcal{D}')). \tag{6.83}$$

Proof. The first inequality is true iff for every $r \in \mathrm{Tupl}(R)$

$$(*_1 \preceq *_2) \otimes (\mathrm{C}_{\mathcal{D}}^{*_1}(\mathcal{D}'))(r) \leq (\mathrm{C}_{\mathcal{D}}^{*_2}(\mathcal{D}'))(r), \tag{6.84}$$

which is indeed the case:

$$(*_1 \preceq *_2) \otimes (\mathrm{C}_{\mathcal{D}}^{*_1}(\mathcal{D}'))(r) =$$

$$
\begin{aligned}
&(*_1 \preceq *_2) \otimes \mathcal{D}(r) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}'(r') \otimes (r' \approx_R r)^{*_1}\big) = \\
&\mathcal{D}(r) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}'(r') \otimes (r' \approx_R r)^{*_1} \otimes (*_1 \preceq *_2)\big) \leq \\
&\mathcal{D}(r) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}'(r') \otimes (r' \approx_R r)^{*_1} \otimes ((r' \approx_R r)^{*_1} \to (r' \approx_R r)^{*_2})\big) \leq \\
&\mathcal{D}(r) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}'(r') \otimes (r' \approx_R r)^{*_2}\big) = \\
&(\mathrm{C}^{*_2}_{\mathcal{D}}(\mathcal{D}'))(r),
\end{aligned}
$$

where we used (2.17) and (2.13). (6.83) is a consequence of (6.82). □

Theorem 75. *Let $*_1, *_2$ be two hedges on $\mathbf{L}$. Then for any RDTs $\mathcal{D}_1$, $\mathcal{D}_2$ and any nonranked RDT $\mathcal{D}$ on R such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$ we have:*

$$(*_1 \preceq *_2) \leq S^{\approx}_{*_1}(\mathcal{D}_1, \mathcal{D}_2) \to S^{\approx}_{*_2}(\mathcal{D}_1, \mathcal{D}_2), \tag{6.85}$$

$$(*_1 \approx *_2) \leq E^{\approx}_{*_1}(\mathcal{D}_1, \mathcal{D}_2) \leftrightarrow E^{\approx}_{*_2}(\mathcal{D}_1, \mathcal{D}_2). \tag{6.86}$$

Proof. Using adjointness (6.85) is equivalent to

$$(*_1 \preceq *_2) \otimes S^{\approx}_{*_1}(\mathcal{D}_1, \mathcal{D}_2) \leq S^{\approx}_{*_2}(\mathcal{D}_1, \mathcal{D}_2).$$

Due to (2.21), (2.15) and (6.82):

$$
\begin{aligned}
&(*_1 \preceq *_2) \otimes S^{\approx}_{*_1}(\mathcal{D}_1, \mathcal{D}_2) = \\
&(*_1 \preceq *_2) \otimes \textstyle\bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \to (\mathrm{C}^{*_1}_{\approx\mathcal{D}}(\mathcal{D}_2))(r')\big) \leq \\
&\textstyle\bigwedge_{r \in \mathrm{Tupl}(R)} \big((*_1 \preceq *_2) \otimes (\mathcal{D}_1(r) \to (\mathrm{C}^{*_1}_{\approx\mathcal{D}}(\mathcal{D}_2))(r'))\big) \leq \\
&\textstyle\bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \to ((*_1 \preceq *_2) \otimes (\mathrm{C}^{*_1}_{\approx\mathcal{D}}(\mathcal{D}_2))(r'))\big) \leq \\
&\textstyle\bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \to (\mathrm{C}^{*_2}_{\approx\mathcal{D}}(\mathcal{D}_2)(r'))\big) = S^{\approx}_{*_2}(\mathcal{D}_1, \mathcal{D}_2).
\end{aligned}
$$

(6.86) is a consequence of (6.85). □

In words, (6.85) says that if $\mathcal{D}_1$ is a subset of $\mathcal{D}_2$ using $*_1$ and if $*_1$ is stronger than $*_2$, then $\mathcal{D}_1$ is a subset of $\mathcal{D}_2$ using $*_2$. Analogously, (6.86) says that if hedges $*_1$ and $*_2$ are similar, then the degree of similarity of $\mathcal{D}_1$ and $\mathcal{D}_2$ using $*_1$ and $*_2$ are similar.

6.4 Conclusions

In this chapter we have investigated the questions related to similarity preservation. We have shown that if a relational operation (from Section 2.3.2) is applied to pairwise similar input arguments (i.e., pairwise similar RDTs), it produces similar results. In addition, the degree of similarity of the results can be estimated based on the degrees of similarity of the input arguments prior to the evaluation of relational operations, i.e., prior to the execution of a relational query. We have also shown that the estimations of similarity are optimal considering all possible RDTs over arbitrary domains with similarities. Later we have investigated similarity of ranked data tables based on pairwise similar tuple values. We have focused on the role of similarity-based closures of ranked data tables which are new and nontrivial relational operations. We have shown that tuple-based similarity can be reduced to rank-based similarity of similarity-based closure.

Shrnutí v českém jazyce

Relační databáze, založené na relačním modelu dat (E. F. Codd 1970 [41]), jsou dnes standardem pro ukládání a manipulaci s daty. Za úspěchem relačního modelu stojí, mimo jiné, jeho pevné matematické základy – teorie množin a (dvouhodnotová) predikátová logika. To, co je na jednu stranu výhodou, je na druhou stranu limitující. Relační databáze založené na klasickém relačním modelu neumí pracovat s koncepty, které nejsou bivalentní, ale vícehodnotové, např. s podobností.

Představme si, že hledáme hotel v Olomouci, který nabízí pokoje za 100 €. Klasické relační databáze nám vrátí množinu hotelů, jejichž cena je přesně 100 €. Je ale přirozené, že vedle hotelů stojících 100 € nás zajímají i hotely, jejichž ceny jsou blízko naší představě (např. hotely s cenou 95 € nebo 105 €). Na úvaze se nic nezmění, budeme-li hledat hotely s cenou v nějakém intervalu, např. 95–105 €. Opět nás budou zcela určitě zajímat i hotely, jejichž cena je dostatečně blízko (je podobná) našim požadavkům, např. hotely s cenou 89 € nebo 110 €.

Snahy rozšířit relační model o podobnosti na doménách (doména je množina možných hodnot pro daný atribut) se objevují už od roku 1982 [31]. Podobnost na doméně D_y atributu y lze formalizovat pomocí binární fuzzy relace $\approx_y$: $D_y \times D_y \to L$. Tedy každým dvěma hodnotám $d_1, d_2 \in D_y$ je přiřazen stupeň jejich podobnosti $(d_1 \approx_y d_2) \in L$. Často se volí $L = [0, 1]$. Relačním modelům, které uvažují podobnosti na doménách, budeme říkat podobnostní relační modely.

Disertační práce je věnována relačnímu modelu dat, který představili Bělohlávek a Vychodil [18], a který rozšiřuje původní relační model takto: 1) na každé doméně je zavedena relace podobnosti, 2) relace (databázové tabulky) jsou rozšířené o tzv. ranky. Každý řádek (záznam) obsahuje navíc rank, což je stupeň, ve kterém daný řádek vyhovuje dotazu. Tento model je založen na predikátové fuzzy logice.

První část disertační práce je zaměřena na funkční závislosti v podobnostních relačních modelech, které se snaží popsat závislosti typu: jestliže jsou si dva řádky podobné na atributech A, pak jsou si podobné na atributech B. Přístupů k funkčním závislostem v podobnostních relačních modelech je několik desítek, pozornost je proto věnována porovnání těchto přístupů. Je představeno kritérium, které umožňuje rozdílné definice objektivně srovnat. U funkčních závislostí, které představili Bělohlávek a Vychodil, jsou A, B fuzzy množiny atributů. Příkladem takové závislosti může být: jestliže mají hotely podobnou cenu alespoň ve stupni 0,8, pak mají podobné hodnocení od zákazníků alespoň ve stupni

0,7. Formálně lze psát $\{^{0,8}/cena\} \Rightarrow \{^{0,7}/\text{hodnocení}\}$. Pravdivost funkčních závislostí se uvažuje ve stupních. V disertační práci je pro tyto funkční závislosti vyvinut alternativní dokazovací systém, který je založen na orientovaných grafech. Je dokázána úplnost v následujícím smyslu: Funkční závislost $A \Rightarrow B$ sémanticky plyne z množiny funkčních závislostí tehdy a jen tehdy, existuje-li orientovaný graf pro $A \Rightarrow B$. Konstrukci orientovaných grafů lze využít i pro určení uzávěru (fuzzy) množiny atributů vzhledem k teorii.

Druhá část práce je věnována citlivosti funkčních závislosti a relačních operací (v modelu Bělohlávka a Vychodila) na vstupních datech. Nejprve je diskutováno, jak lze měřit podobnost databázových tabulek (relací s ranky) a jsou představeny dvě míry: podobnost založená na rancích (rank-based similarity) a podobnost založená na datech (tuple-based similarity). U podobnosti založené na rancích řekneme, že dvě relace s ranky jsou si podobné, pokud stejné řádky patří do obou relací v podobném stupni. Pro tuto podobnost je dokázáno, že v podobných relacích budou funkční závislosti platit v podobném stupni. Tedy, že definice funkčních závislostí je robustní: malá změna na vstupních datech způsobí pouze malou změnu v platnosti funkčních závislostí. Rovněž jsou prezentovány odhady pro pravdivost funkční závislosti $A_1 \Rightarrow B_1$, $A_2 \Rightarrow B_2$ v závislosti na podobnosti fuzzy množin atributů A_1, A_2 a B_1, B_2. Pro podobnost založenou na rancích je dále studována citlivost výsledků relačních operací na vstupních datech. Je ukázáno, že pro libovolný dotaz lze podobnost výsledků dotazu odhadnout na základě podobnosti vstupních dat.

U podobnosti založené na datech řekneme, že dvě relace s ranky jsou si podobné, jestliže ke každému řádku v jedné relaci existuje řádek v druhé relaci, který je mu podobný a opačně. Ukazuje se, že tuto podobnost lze vyjádřit pomocí podobnosti založené na rancích a nové relační operace: podobnostního uzávěru. V disertační práci jsou studovány vlastnosti podobnostního uzávěru a jeho vztah k relačním operacím. Rovněž je představena podobnost pro relace s ranky, která zobecňuje obě předchozí.

Bibliography

[1] W. W. Armstrong. Dependency structures of data base relationships. In Jack L. Rosenfeld and Herbert Freeman, editors, *Information Processing 74: Proceedings of IFIP Congress*, pages 580–583, Amsterdam, 1974. North Holland.

[2] A. Aussem and J. M. Petit. e-functional dependency inference: application to dna microarray expression data. In *BDA*, 2002.

[3] F. E. Petry B. P. Buckles and H. S. Sachar. Design of similarity-based relational databases. In Constantin V. Negoita Henri Prade, editor, *Fuzzy logic in knowledge engineering*, pages 3–17. TUV Rheinland, 1986.

[4] J.F. Baldwin and S.Q Zhou. A fuzzy relational inference language. *Fuzzy Sets and Systems*, 14(2):155 – 174, 1984.

[5] J. Bang-Jensen and G. Gutin. *Digraphs: Theory, Algorithms and Applications*. Springer Monographs in Mathematics. Springer, 2010.

[6] C. Beeri and P. A. Bernstein. Computational problems related to the design of normal form relational schemas. *ACM Trans. Database Syst.*, 4:30–59, March 1979.

[7] C. Beeri, R. Fagin, and J. H. Howard. A complete axiomatization for functional and multivalued dependencies in database relations. In *Proceedings of the 1977 ACM SIGMOD international conference on Management of data*, SIGMOD '77, pages 47–61, New York, NY, USA, 1977. ACM.

[8] R. Belohlavek. Fuzzy closure operators. *Journal of Mathematical Analysis and Applications*, 262(2):473 – 489, 2001.

[9] R. Belohlavek. *Fuzzy Relational Systems: Foundations and Principles*. Kluwer Academic Publishers, 2002.

[10] R. Belohlavek. Fuzzy closure operators induced by similarity. *Fundamenta Informaticae*, 58:79–91, 2003.

[11] R. Belohlavek, L. Urbanova, and V. Vychodil. Similarity of query results in similarity-based databases. In JingTao Yao, Sheela Ramanna, Guoyin Wang, and Zbigniew Suraj, editors, *Rough Sets and Knowledge Technology*, volume 6954 of *Lecture Notes in Computer Science*, pages 258–267. Springer Berlin Heidelberg, 2011.

[12] R. Belohlavek, L. Urbanova, and V. Vychodil. Sensitivity analysis for declarative relational query languages with ordinal ranks. In H. Tompits, S. Abreu, J. Oetsch, J. Pührer, D. Seipel, M. Umeda, and A. Wolf, editors, *Applications of Declarative Programming and Knowledge Management*, volume 7773 of *Lecture Notes in Computer Science*, pages 58–76. Springer Berlin Heidelberg, 2013.

[13] R. Belohlavek and V. Vychodil. Attribute implications in a fuzzy setting. In R. Missaoui and J. Schmidt, editors, *Formal Concept Analysis*, volume 3874 of *Lecture Notes in Computer Science*, pages 45–60. Springer Berlin / Heidelberg, 2006.

[14] R. Belohlavek and V. Vychodil. Axiomatization of fuzzy attribute logic over complete residuated lattices. In *Proceedings of the 2006 Joint Conference on Information Sciences, JCIS 2006, Kaohsiung, Taiwan, ROC, October 8-11*, 2006.

[15] R. Belohlavek and V. Vychodil. Data tables with similarity relations: Functional dependencies, complete rules and non-redundant bases. In *Proceedings of the 11th International Conference on Database Systems for Advanced Applications*, DASFAA'06, pages 644–658, Berlin, Heidelberg, 2006. Springer-Verlag.

[16] R. Belohlavek and V. Vychodil. Similarity issues in attribute implications from data with fuzzy attributes. In *Information Reuse and Integration, 2006 IEEE International Conference on*, pages 132–135, Sept 2006.

[17] R. Belohlavek and V. Vychodil. On proofs and rule of multiplication in fuzzy attribute logic. In P. Melin, O. Castillo, L. Aguilar, J. Kacprzyk, and W. Pedrycz, editors, *Foundations of Fuzzy Logic and Soft Computing*, volume 4529 of *Lecture Notes in Computer Science*, pages 471–480. Springer Berlin / Heidelberg, 2007.

[18] R. Belohlavek and V. Vychodil. Data dependencies in codd's relational model with similarities. In José Galindo, editor, *Handbook of Research on Fuzzy Information Processing in Databases*, pages 634–657. IGI Global, 2008.

[19] R. Belohlavek and V. Vychodil. Logical foundations for similarity-based databases. In Lei Chen, Chengfei Liu, Qing Liu, and Ke Deng, editors, *Database Systems for Advanced Applications*, volume 5667 of *Lecture Notes in Computer Science*, pages 137–151. Springer Berlin Heidelberg, 2009.

[20] R. Belohlavek and V. Vychodil. Query systems in similarity-based databases: Logical foundations, expressive power, and completeness. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1648–1655, New York, NY, USA, 2010. ACM.

[21] R. Belohlavek and V. Vychodil. Codd's relational model from the point of view of fuzzy logic. *Journal of Logic and Computation*, 21:851–862, 2011.

[22] R. Belohlavek and V. Vychodil. Relational algebra for multi-ranked similarity-based databases. In *IEEE Symposium on Foundations of Computational Intelligence, FOCI 2013, Singapore, Singapore, April 16-19, 2013*, pages 1–8. IEEE, 2013.

[23] R. Belohlavek and V. Vychodil. Relational similarity-based databases, part 1: Foundations and query systems. *Submitted*, 2014.

[24] R. Belohlavek and V. Vychodil. Relational similarity-based databases, part 2: Dependencies in data. *Submitted*, 2014.

[25] F. Berzal, I. Blanco, D. Sánchez, J. M. Serrano, and M. A. Vila. A definition for fuzzy approximate dependencies. *Fuzzy Sets Syst.*, 149(1):105–129, January 2005.

[26] B. Bhuniya and P. Niyogi. Lossless join property in fuzzy relational databases. *Data & Knowledge Engineering*, 11(2):109–124, 1993.

[27] P. Bosc, D. Dubois, and H. Prade. Fuzzy functional dependencies - an overview and a critical discussion. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems*, pages 325–330, 1994.

[28] P. Bosc, D. Dubois, and H. Prade. Fuzzy functional dependencies and redundancy elimination. *J. Am. Soc. Inf. Sci.*, 49:217–235, March 1998.

[29] P. Bosc, L. Lietard, and O. Pivert. Extended functional dependencies as a basis for linguistic summaries. In *Proceedings of the Second European Symposium on Principles of Data Mining and Knowledge Discovery*, PKDD '98, pages 255–263, London, UK, UK, 1998. Springer-Verlag.

[30] P. Bosc, O. Pivert, and L. Ughetto. Database mining for the discovery of extended functional dependencies. In *Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American*, pages 580 –584, jul 1999.

[31] B. P. Buckles and F. E. Petry. A fuzzy representation of data for relational databases. *Fuzzy Sets and Systems*, 7(3):213 – 226, 1982.

[32] B. P. Buckles and F. E. Petry. Extending the fuzzy database with fuzzy numbers. *Information Sciences*, 34(2):145 – 155, 1984.

[33] B. P. Buckles and F. E. Petry. Uncertainty models in information and database systems. *Journal of Information Science*, 11(2):77 – 87, 1985.

[34] P. Budikova, M. Batko, and P. Zezula. Query language for complex similarity queries. In T. Morzy, T. Harder, and R. Wrembel, editors, *Advances in Databases and Information Systems*, volume 7503 of *Lecture Notes in Computer Science*, pages 85–98. Springer Berlin Heidelberg, 2012.

[35] G. Chen. Fuzzy functional dependencies and a series of design issues of fuzzy relational databases. In *Fuzzines in Database Management Systems*, pages 166–185. Physica Verlag, Heidelberg, 1995.

[36] G. Chen, E. E. Kerre, and J. Vandenbulcke. A computational algorithm for the ffd transitive closure and a complete axiomatization of fuzzy functional dependencies. *International Journal of Intelligent Systems*, 9:421–439, 1994.

[37] G. Chen, E. E. Kerre, and J. Vandenbulcke. An extended boyce-codd normal form in fuzzy relational databases. In *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, volume 3, pages 1546–1551 vol.3, Sep 1996.

[38] G. Chen, E. E. Kerre, and J. Vandenbulcke. Normalization based on fuzzy functional dependency in a fuzzy relational data model. *Information Systems*, 21:299–310, 1996.

[39] G.Q. Chen. A step towards the theory of fuzzy relational database design. In *Proc. of IFSA'91 World Congress*, pages 44–47, 1991.

[40] R. Cignoli, F. Esteva, L. Godo, and A. Torrens. Basic fuzzy logic is the logic of continuous t-norms and their residua. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 4:106–112, 2000.

[41] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, June 1970.

[42] E. F. Codd. Extending the database relational model to capture more meaning. *ACM Trans. Database Syst.*, 4(4):397–434, December 1979.

[43] E. F. Codd. More commentary on missing information in relational databases (applicable and inapplicable information). *SIGMOD Rec.*, 16(1):42–50, March 1987.

[44] P. Cordero, M. Enciso, A. Mora, and I. Perez de Guzman. A complete axiomatic system for fuzzy functional dependencies over domains with similarity relations. *Lecture Notes Computer Science - IWANN 09*, 5517:261–269, 2009.

[45] P. Cordero, M. Enciso, A. Mora, I. Perez de Guzman, and J.M. Rodriguez-Jimenez. Specification and inference of fuzzy attributes. In *Foundations of Computational Intelligence (FOCI), 2011 IEEE Symposium on*, pages 107–114, 2011.

[46] J. C. Cubero, J. M. Medina, O. Pons, and M. A. Vila. Non-transitive fuzzy dependencies (i). *Fuzzy Sets Syst.*, 106:401–431, September 1999.

[47] J. C. Cubero, J. M. Medina, O. Pons, and M. A. Vila. Transitive fuzzy dependencies (ii). *Fuzzy Sets Syst.*, 106:433–448, September 1999.

[48] J. C. Cubero and M. A. Vila. A new definition of fuzzy functional dependency in fuzzy relational databases. *International Journal of Intelligent Systems*, 9(5):441–448, 1994.

[49] T. H. Dang and D. K. Tran. Comments on fuzzy data dependencies and implication of fuzzy data dependencies. *Fuzzy Sets and Systems*, 148(1):153–156, 2004. Web Mining Using Soft Computing.

[50] C. J. Date. *Relational Database:Selected Writings*. Addison Wesley Publishing Company, 1986.

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

[52] C. J. Date. *Date on Database: Writings 2000–2006*. Apress, 2006.

[53] C. J. Date. *Database Design and Relational Theory: Normal Forms and All That Jazz*. O'Reilly Media; 1 edition, 2012.

[54] C. J. Date and H. Darwen. *Databases, Types and the Relational Model (3rd Edition)*. Addison-Wesley, 2006.

[55] M. Demirci. Fuzzy functions and their applications. *Journal of Mathematical Analysis and Applications*, 252(1):495 – 517, 2000.

[56] D. Dubois and H. Prade. Certainty and uncertainty of (vague) knowledge and generalized dependencies in fuzzy databases. In *Fuzzy Engineering Toward Human Friendly Systems*, pages 239–249. IOS Press, 1992.

[57] D. Dubois and H. Prade. Gradualness, uncertainty and bipolarity: Making sense of fuzzy sets. *Fuzzy Sets and Systems*, 192(0):3 – 24, 2012. Fuzzy Set Theory — Where Do We Stand and Where Do We Go?

[58] F. Esteva, L. Godo, and C. Noguera. A logical approach to fuzzy truth hedges. *Information Sciences*, 232:366–385, 2013.

[59] R. Fagin. Fuzzy queries in multimedia database systems. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '98, pages 1–10, New York, NY, USA, 1998. ACM.

[60] W. Fan, H. Gao, X. Jia, J. Li, and S. Ma. Dynamic constraints for record matching. *The VLDB Journal*, 20(4):495–520, August 2011.

[61] J. Vandenbulcke G. Q. Chen and E. E. Kerre. Fuzzy functional dependency and its axiomatic system in a fuzzy relational data model. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty (IPMU)*, pages 313–316, 1992.

[62] B. Ganter and R. Wille. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1997.

[63] S. Gottwald. Fuzzy uniqueness of fuzzy mappings. *Fuzzy Sets and Systems*, 3(1):49 – 74, 1980.

[64] S. Gottwald. Mathematical fuzzy logics. *Bulletin of Symbolic Logic*, 14:210–239, 6 2008.

[65] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *Proceedings of the Twenty-sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '07, pages 31–40, New York, NY, USA, 2007. ACM.

[66] M. Hajdinjak and G. Bierman. Extending relational algebra with similarities. *Mathematical. Structures in Comp. Sci.*, 22(4):686–718, August 2012.

[67] P. Hajek. *Metamathematics of Fuzzy Logic*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[68] P. Hajek. On very true. *Fuzzy Sets and Systems*, 124(3):329–333, 2001.

[69] R. Holzer. Knowledge acquisition under incomplete knowledge using methods from formal concept analysis: Part I. *Fundam. Inform*, 63(1):17–39, 2004.

[70] L. Jezkova, P. Cordero, and M. Enciso. Codd's relational model of data over domains with similarities: A comparative survey. *Submitted*, 2014.

[71] A. Kiss. λ decomposition of fuzzy relational databases. *Annales Univ. Sci. Budapest*, 12:133–142, 1991.

[72] F. Klawonn. Fuzzy points, fuzzy relations and fuzzy functions. In Vilém Novák and Irina Perfilieva, editors, *Discovering the World with Fuzzy Logic*, pages 431–453. Physica-Verlag GmbH, Heidelberg, Germany, Germany, 2000.

[73] F. Klawonn and J. L. Castro. Similarity in fuzzy reasoning. *Mathware & Soft Computing*, 2:197–228, 1995.

[74] P. Krajca and V. Vychodil. Foundations of relational similarity-based query language RESIQL. In *IEEE Symposium on Foundations of Computational Intelligence, FOCI 2013, Singapore, Singapore, April 16-19, 2013*, pages 15–23, 2013.

[75] W. H. Lee and C. T. Pang. An extension of semantic proximity for fuzzy functional dependencies. In *The 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS2009)*, 2009.

[76] J.Y. C. Liu and C. H. Huang. Handling missing data in extended possibility-based fuzzy relational databases. In *Innovations in Bio-Inspired Computing and Applications (IBICA), 2012 Third International Conference on*, pages 57 –62, sept. 2012.

[77] W. Y. Liu. Extending the relational model to deal with fuzzy values. *Fuzzy Sets Syst.*, 60:207–212, December 1993.

[78] W. Y. Liu. Constraints on fuzzy values and fuzzy functional dependencies. *Information Sciences*, 78(3-4):303–309, 1994.

[79] W. Y. Liu. Fuzzy data dependencies and implication of fuzzy data dependencies. *Fuzzy Sets Syst.*, 92:341–348, December 1997.

[80] Z. M. Ma, W. J. Zhang, W. Y. Ma, and F. Mili. Data dependencies in extended possibility-based fuzzy relational databases. *International Journal of Intelligent Systems*, 17(3):321–332, 2002.

[81] D. Maier. Minimum covers in relational database model. *Journal of the ACM*, 27(4):664–674, October 1980.

[82] D. Maier. *Theory of Relational Databases.* Computer Science Pr, Rockville, MD, USA, 1983.

[83] J. M. Medina, M. A. Vila, J. C. Cubero, and O. Pons. Towards the implementation of a generalized fuzzy relational database model. *Fuzzy Sets Syst.*, 75:273–289, November 1995.

[84] A. Melton and S. Shenoi. Fuzzy relations and fuzzy relational databases. *Computers & Mathematics with Applications*, 21(11-12):129–138, 1991.

[85] N. Mouaddib and N. Bonanno. New semantics for the membership degree in fuzzy databases. In *Uncertainty Modeling and Analysis, 1995, and Annual Conference of the North American Fuzzy Information Processing Society. Proceedings of ISUMA - NAFIPS '95., Third International Symposium on*, pages 655 –660, sep 1995.

[86] K. Myszkorowski. Analysis of fuzzy n-ary relations with the use of interval-valued fuzzy functional dependencies. *International Journal of General Systems*, 42, 2013.

[87] M. Nakata. Dependencies in fuzzy databases: functional dependency. In *Proceedings of 1995 IEEE International Conference on Fuzzy Systems*, volume 2, pages 757–764, Yokohama, Japan, 1995.

[88] L. Lietard P. Bosc and O. Pivert. Functional dependencies revisited under graduality and imprecision. In *Fuzzy Information Processing Society, 1997. NAFIPS '97., 1997 Annual Meeting of the North American*, pages 57–62, 1997.

[89] J. Pavelka. On fuzzy logic I: Many-valued rules of inference. *Mathematical Logic Quarterly*, 25(3–6):45–52, 1979.

[90] J. Pavelka. On fuzzy logic II: Enriched residuated lattices and semantics of propositional calculi. *Mathematical Logic Quarterly*, 25(7–12):119–134, 1979.

[91] J. Pavelka. On fuzzy logic III: Semantical completeness of some many-valued propositional calculi. *Mathematical Logic Quarterly*, 25(25–29):447–464, 1979.

[92] H. Prade. Lipski's approach to incomplete information data bases restated and generalized in the setting of zadeh's possibility theory. *Information Systems*, 9(1):27–42, 1984.

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

[94] K. V. S. V. N. Raju and A. K. Majumdar. The study of joins in fuzzy relational databases. *Fuzzy Sets Syst.*, 21(1):19–34, January 1987.

[95] 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 Syst.*, pages 129–166, 1988.

[96] D. Rasmussen and R. R. Yager. Finding fuzzy and gradual functional dependencies with summarysql. *Fuzzy Sets and Systems*, 106(2):131 – 142, 1999.

[97] A. N. Saharia and T. M. Barron. Approximate dependencies in database systems. *Decision Support Systems*, 13(3-4):335–347, March 1995.

[98] P. C. Saxena and B. K. Tyagi. Fuzzy functional dependencies and independencies in extended fuzzy relational database models. *Fuzzy Sets and Systems*, 69(1):65–89, 1995.

[99] A.K. Sharma, A. Goswami, and D.K. Gupta. Fuzzy inclusion dependencies in fuzzy relational databases. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 1, pages 507 – 510 Vol.1, april 2004.

[100] S. Shenoi and A. Melton. An extended version of the fuzzy relational database model. *Information Sciences*, 52:35–52, 1990.

[101] S. Shenoi, A. Melton, and L. T. Fan. An equivalence classes model of fuzzy relational databases. *Fuzzy Sets and Systems*, 38(2):153–170, 1990.

[102] M. Shirvanian and W. Lippe. Optimization of the normalization of fuzzy relational databases by using alternative methods of calculation for the fuzzy functional dependency. In *2008 IEEE International Conference on Fuzzy Systems*, pages 15–20, 2008.

[103] M. I. Sozat and A. Yazici. A complete axiomatization for fuzzy functional and multivalued dependencies in fuzzy database relations. *Fuzzy Sets and Systems*, 117:161–181, 2001.

[104] Y. Takahashi. Fuzzy database query languages and their relational completeness theorem. *Knowledge and Data Engineering, IEEE Transactions on*, 5(1):122–125, Feb 1993.

[105] G. Takeuti and S. Titani. Globalization of intuitionistic set theory. *Annals of Pure and Applied Logic*, 33:195–211, 1990.

[106] B. K. Tyagi, A. Sharfuddin, R. N. Dutta, and D. K. Tayal. A complete axiomatization of fuzzy functional dependencies using fuzzy function. *Fuzzy Sets and Systems*, 151:363–379, 2005.

[107] M. Umano. Freedom-o: A fuzzy database system. In Gupta Sanchez, editor, *Fuzzy Information and Decision Processes*, pages 339–347. North-Holand Pub. Comp., 1982.

[108] M. Umano. Retrieval from fuzzy databases by fuzzy relational algebra. In G. Sanchez, editor, *Fuzzy Information Knowledge Representation and Decision Analysis*, pages 1–6. Pergamon Press, Oxford, 1983.

[109] L. Urbanova and V. Vychodil. Derivation digraphs for dependencies in ordinal and similarity-based data. *Information Sciences*, 268(0):381 – 396, 2014. New Sensing and Processing Technologies for Hand-based Biometrics Authentication.

[110] L. Urbanova, V. Vychodil, and L. Wiese. Applications of ordinal ranks to flexible query answering. In *Scalable Uncertainty Management - 6th International Conference, SUM 2012, Marburg, Germany, September 17-19, 2012. Proceedings*, pages 16–29, 2012.

[111] M. Vucetic and M. Vujosevic. A literature overview of functional dependencies in fuzzy relational database models. *Technics Technologies Education Management-TTEM*, 7(4):1593–1604, 2012.

[112] S. L. Wang, J. W. Shen, and T. P. Hong. Mining fuzzy functional dependencies from quantitative data. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, volume 5, pages 3600 –3605 vol.5, 2000.

[113] S. L. Wang, J. S. Tsai, and B. C. Chien. Mining approximate dependencies using partitions on similarity-relation-based fuzzy databases. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 5, pages 871 –875 vol.5, 1999.

[114] S. B. Yahia and A. Jaoua. Mining linguistic summaries of databases using based łukasiewicz implication fuzzy functional dependency. In *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE '99. 1999 IEEE International*, volume 3, pages 1246 –1250 vol.3, aug. 1999.

[115] S. Ben Yahia, H. Ounalli, and A. Jaoua. An extension of classical functional dependency: dynamic fuzzy functional dependency. *Information Sciences*, 119(3-4):219 – 234, 1999.

[116] A. Yazici, E. Gocmen, B.P. Buckles, R. George, and F.E. Petry. An integrity constraint for a fuzzy relational database. In *Proc. of Second IEEE Int. Conf. on Fuzzy Systems 1*, pages 496–499, 1993.

[117] A. Yazici and M.I. Sozat. The integrity constraints for similarity-based fuzzy relational databases. *International Journal of Intelligent Systems*, 13:641–659, 1998.

[118] L. A. Zadeh. Fuzzy sets. *Information and control*, 8:338 – 353, 1965.

[119] L. A. Zadeh. Similarity relations and fuzzy orderings. *Inf. Sci.*, 3(2):177–200, April 1971.

[120] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3 – 28, 1978.

[121] L.A. Zadeh. Pruf—a meaning representation language for natural languages. *International Journal of Man-Machine Studies*, 10(4):395 – 460, 1978.

[122] F. Zhao and Z.M. Ma. Functional dependencies in vague relational databases. In *2006 IEEE International Conference on Systems, Man, and Cybernetics*, pages 4006–4010, 2006.

[123] A. Zvieli. A fuzzy relational calculus. In *Expert Database Conf.'86*, pages 311–326, 1986.

Přírodovědecká fakulta

Univerzity Palackého v Olomouci

Katedra informatiky

The role of similarity in relational databases

Autoreferát disertační práce k získání
akademicko-vědeckého titulu Ph.D.

Lucie Ježková

2015

Uchazeč: Mgr. Lucie Ježková

Školitel: doc. RNDr. Vilém Vychodil, Ph.D.

Oponenti:

S disertační prací a posudky se bude možné seznámit na katedře informatiky PřF UP, 17. listopadu 12, 771 46 Olomouc.

Contents

1 Problem setting **3**
- 1.1 Introduction . . . 3
- 1.2 Preliminaries . . . 4
  - 1.2.1 Residuated lattices . . . 4
  - 1.2.2 The relational model . . . 6
  - 1.2.3 Ranked data tables over domains with similarities . . . 7

2 Overview of similarity-based functional dependencies **12**
- 2.1 Generalizations of the relational model . . . 13
- 2.2 Comparison of similarity-based generalizations of FD . . . 14

3 Derivation digraphs for graded if-then rules **20**
- 3.1 Derivation acyclic digraphs for FAIs . . . 20
- 3.2 Completeness . . . 22
- 3.3 Computing closures . . . 23

4 Sensitivity analysis for similarity-based functional dependencies **25**
- 4.1 Rank-based similarity . . . 25
- 4.2 Similarity estimates for FD . . . 26

5 Similarity estimates of query results **28**
- 5.1 Similarity estimates for relational operations . . . 28
- 5.2 Similarity of complex query results . . . 32
- 5.3 Tuple-based similarity . . . 34
  - 5.3.1 Similarity-based semijoins and closures . . . 35
  - 5.3.2 Tuple-based similarity estimates . . . 36
  - 5.3.3 Unifying approach to similarity of RDTs . . . 37

6 Conclusions **38**

Summary in Czech language **39**

Selected publications of the author **41**

Chapter 1

Problem setting

1.1 Introduction

Similarity search and related issues are current topic in databases. Over the last ten years, several thousands papers dealing with similarity in databases were published according to Web of Science. We study a particular rank-aware relational model over domains with similarities, which was introduced by Belohlavek and Vychodil [12, 13, 14, 17, 18].

Belohlavek and Vychodil extended the Codd's orignal model in the following way: domains are additionally equipped with similarity relations and each tuple in a data table has assigned a rank. The rank is a degree to which a tuple matches a similarity-based query. Both similarity degrees and ranks come from complete residuated lattice. We will investigate the model proposed by Belohlavek and Vychodil. First, we study similarity-based functional dependencies (SBFDs). We propose a graph-based method for reasoning and show a correspondence between construction of a directed graph and normalized proof. We also compare the definition of SBFD given by Belohlavek and Vychodil with other approaches. Second, we examine sensitivity issues. We define two similarity measures for ranked-data tables (RDTs): ranked-based similarity (two RDTs are considered similar if they contain tuples with similar ranks) and tuple-based similarity (two RDTs are considered similar if they contain tuples with similar values). The tuple-based similarity can be expressed by rank-based similarity and a new relational operation, called similarity-based closure. Using the notion of rank-based similarity, we show that the similarity of query results can be estimated based on the similarity of input data prior to query execution. Such estimates can be provided for arbitrary complex queries. We further study estimates for tuple-based similarity and properties of the similarity-based closure. We also explore sensitivity issues connected to similarity-based functional dependencies.
The document is organized as follows:
In Chapter 1.2 we summarize basic facts from residuated lattices, fuzzy set theory and Codd's relational model of data. We also introduce the model proposed by Belohlavek and Vychodil.

In Chapter 2 we review and critically examine the existing work on similarity-based functional dependencies. We try to objectively compare various approaches and we propose a novel criterion to achieve this goal.

In Chapter 3 we show that degrees to which a SBFD semantically follows from sets (or

graded sets) of other SBFDs can be characterized by existence of particular directed acyclic graphs with vertices labeled by attributes and degrees coming from complete residuated lattices. In addition, we show that the construction of directed acyclic graphs can be used to compute closures of sets of attributes.

In Chapter 4 we define the rank-based similarity of RDTs and show that a SBFD holds in similar data tables to similar degree. We also explore how the validity of SBFD change if we replace the antecedent (or consequent) by similar set of attributes.

In Chapter 5 we show that relational operations preserve the rank-based similarity of RDTS. We also provide an alternative definition of similarity of RDTs (tuple-based similarity) and explore its preservation for relational operations. The tuple-based similarity is closely related to a new relational operation, a similarity-based closure, whose properties are investigated as well. We also outline a general approach to similarity of RDTs that includes both the rank-based similarity and tuple-based similarity.

1.2 Preliminaries

In this section we recall the basic facts of residuated lattices, fuzzy set theory, and relational model of data. We also introduce one extension of the Codd's model of data, namely ranked data tables over domains with similarities.

1.2.1 Residuated lattices

A complete residuated lattice [4, 50], which will serve as a basic structure of truth degrees, is an algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \to, 0, 1\rangle$ such that

- $\langle L, \wedge, \vee, 0, 1\rangle$ is a complete lattice with 0 and 1 being the least and the greatest element of L;
- $\langle L, \otimes, 1\rangle$ is a commutative monoid, i.e. $\otimes$ is a binary operation which is commutative, associative, and $a \otimes 1 = 1 \otimes a = a$ for each $a \in L$;
- $\otimes$ and $\to$ satisfy so-called adjointness property:

$$a \otimes b \leq c \text{ iff } a \leq b \to c \tag{1.1}$$

  for each $a, b, c \in L$, where $\leq$ is the order induced by the lattice structure of $\mathbf{L}$, i.e. $a \leq b$ iff $a = a \wedge b$.

Elements a of L are interpreted as truth degrees. The operations $\otimes$ and $\to$ are truth functions of "fuzzy conjunction" and "fuzzy implication" and are called a multiplication and a residuum, respectively. For a complete residuated lattice $\mathbf{L}$ we define

$$a \leftrightarrow b = (a \to b) \wedge (b \to a) \tag{1.2}$$

and call this derived operation a biresiduum. The biresiduum can be seen as a truth function for an equivalence. For a nonnegative integer n, the n-th power of $a \in L$ is defined by

$$a^0 = 1 \quad \text{and} \quad a^{n+1} = a^n \otimes a. \tag{1.3}$$

The unit interval: Common examples of complete residuated lattices include structures defined on the real unit interval, i.e. structures $\mathbf{L}$ where $L = [0, 1]$, $\wedge$ and $\vee$ being minimum and maximum, respectively, and $\otimes$ being a left-continuous triangular norm (shortly, a t-norm) with the corresponding $\rightarrow$. All complete residuated lattices on the real unit interval with continuous $\otimes$ can be constructed by means of ordinal sums [29] from the following three pairs of adjoint operations:

Łukasiewicz:	$a \otimes b = \max(a + b - 1, 0)$,	$a \rightarrow b = \min(1 - a + b, 1)$;
Gödel:	$a \otimes b = \min(a, b)$,	$a \rightarrow b = b$ if $a > b$, 1 otherwise;
Goguen:	$a \otimes b = a \cdot b$,	$a \rightarrow b = \frac{b}{a}$ if $a > b$, 1 otherwise.

Complete resitudated lattices $\langle [0, 1], \min, \max, \otimes, \rightarrow, 0, 1\rangle$ with universe $[0, 1]$ and with Łukasiewicz, Gödel or Goguen operations will be called standard Łukasiewicz, Gödel and Goguen algebra, respectively, and will be denoted as $[0, 1]_{\mathrm{L}}, [0, 1]_G, [0, 1]_{\Pi}$. Sometimes we will denote $\rightarrow_{\mathrm{L}}$, $\rightarrow_G$, and $\rightarrow_{\Pi}$ to emphasize the Łukasiewicz, Gödel and Goguen implication, respectively.

We now turn our attention to unary operations called truth-stressing hedges [51, 50, 42]. Let $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ be a complete residuated lattice. An unary operation $^* : L \rightarrow L$ satisfying

$$1^* = 1, \tag{1.4}$$
$$a^* \leq a, \tag{1.5}$$
$$(a \rightarrow b)^* \leq a^* \rightarrow b^*, \tag{1.6}$$
$$a^{**} = a^*, \tag{1.7}$$

for each $a, b \in L$ will be called a truth-stressing hedge (or shortly hedge) for $\mathbf{L}$. The algebra $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, ^*, 0, 1\rangle$ is then called a complete residuated lattice with hedge and denoted as $\mathbf{L^*}$. Hedge * can be understood as a truth function of unary connective "very true". If ϕ is a proposition with truth degree $||\phi||$, then the truth degree of proposition "ϕ is very true" (or "it is very true that ϕ") is $||\phi||^*$. Properties (1.4)–(1.7) have natural interpretations, e.g., (1.5) can be read: "if a is very true, then a is true".

Two boundary cases of (truth-stressing) hedges are

(i) identity, i.e., $a^* = a$ $(a \in L)$;

(ii) globalization [82]:

$$a^* = \begin{cases} 1, & \text{if } a = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{1.8}$$

If * is a globalization, then $(a \rightarrow b)^* = 1$ iff $a \rightarrow b = 1$ iff $a \leq b$.

Since * is intensive (1.5), monotone (consequence of (1.4) and (1.6)) and idempotent (1.7), it is an interior operator. We may therefore denote by $\mathrm{fix}(*)$ the set of all fixed points:

$$\mathrm{fix}(*) = \{a \in L \mid a^* = a\} = \{a^* \mid a \in L\}. \tag{1.9}$$

If $*_1$ and $*_2$ are two hedges on $\mathbf{L}$ such that $\mathrm{fix}(*_1) \subseteq \mathrm{fix}(*_2)$ we say that $*_1$ is stronger than $*_2$.

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 **2**, which is the structure of truth degrees of the classical logic. That is, the operations $\wedge,\vee,\otimes,\rightarrow$ of **2** are the truth functions of the corresponding logical connectives of the classical logic and $0^*=0$, $1^*=1$.

L-sets and L-relations [4, 93]

An **L**-set (or fuzzy set) in a universe X is a mapping $A: X \rightarrow L$, where L is a support of a complete residuated lattice **L**. The degree $A(x)$ is interpreted as a degree to which an element x belongs to A. The set of all **L**-sets in X is denoted by L^X. We are going to use the following notation for denoting **L**-sets: If $X=\{x_1,\ldots,x_n\}$ then an **L**-set A in X can be denoted by $A=\{{}^{a_1}/_{x_1},\ldots,{}^{a_n}/_{x_n}\}$ meaning that $A(x_i)$ equals a_i for each $i=1,\ldots,n$. Operations with **L**-sets are defined component-wise, for $A,B\in L^X$ we have:

$$(A\cup B)(u)=A(u)\vee B(u), \tag{1.10}$$

$$(A\cap B)(u)=A(u)\wedge B(u), \tag{1.11}$$

$$(A\otimes B)(u)=A(u)\otimes B(u), \tag{1.12}$$

$$(A\rightarrow B)(u)=A(u)\rightarrow B(u). \tag{1.13}$$

For **L**-sets $A,B\in L^X$ we define a degree of subsethood of A in B and a degree of equality of A,B as follows:

$$S(A,B)=\bigwedge_{x\in X}(A(x)\rightarrow B(x)), \tag{1.14}$$

$$E(A,B)=\bigwedge_{x\in X}(A(x)\leftrightarrow B(x)). \tag{1.15}$$

The subsethood relation (1.14) generalizes the classical subsethood relation "$\subseteq$". In particular, we have $S(A,B)=1$ (A is fully included in B) iff $A(x)\leq B(x)$ for each $x\in X$. $S(A,B)$ can be understood as a truth degree of the following formula: "for every $x\in X$: if x belongs to A, then x belongs to B." And similarly $E(A,B)$ can be thought of as a truth degree of the formula "for every $x\in X$: x belongs to A iff x belongs to B."

An n-ary **L**-relation between sets $X_1,\ldots,X_n$ is an **L**-set $I\in L^{X_1\times\ldots\times X_n}$. Thus a binary **L**-relation on X is a mapping $I: X\times X\rightarrow L$ that assigns to each pair of elements $x,y\in X$ a degree to which they are related according to I. A binary **L**-relation I on X is called an **L**-equivalence if it is reflexive, symmetric and $\otimes$-transitive, that is for all $x,y,z\in X$:

$$I(x,x)=1, \tag{1.16}$$

$$I(x,y)=I(y,x), \tag{1.17}$$

$$I(x,y)\otimes I(y,z)\leq I(x,z). \tag{1.18}$$

L-equivalence, or fuzzy equivalence, will be denoted as $\equiv$. Note that (1.15) is an **L**-equivalence on L^X. A binary **L**-relation I that is reflexive and symmetric will be called similarity and denoted as $\approx$. We will write $x\approx y$ instead of $\approx(x,y)$. An **L**-equivalence that satisfies separability $I(x,y)=1$ iff $x=y$ will be called **L**-equality.

1.2.2 The relational model

Now we will present the basic notions from the relational model of data, which was introduced by Codd in [30]. For further details see [62, 38]. Let Y denotes a set of attribute

names. For each attribute $y \in Y$ we consider its domain D_y, which is an arbitrary nonempty set of all values allowed for y. A relation scheme is a finite subset $R \subseteq Y$. In particular $R = \emptyset \subseteq Y$ is an empty relation scheme. For each relation scheme R, $\mathrm{Tupl}(R)$ denotes $\prod_{y \in R} D_y$, i.e. the Cartesian product of domains D_y ($y \in R$). Recall that the Cartesian product is a set of all maps $r\colon R \to \bigcup_{y \in R} D_y$ such that $r(y) \in D_y$ holds for all $y \in R$. For $R = \emptyset$ we get $\prod_{y \in \emptyset} D_y = \{\emptyset\}$. A data table $\mathcal{D}$ on R is any finite subset of $\mathrm{Tupl}(R)$. Each $r \in \mathrm{Tupl}(R)$ is called a *tuple over* R and $r(y)$ is called the *y-value of* r. The only data tables on relation scheme $R = \emptyset$ are $\mathcal{D}_\top = \{\emptyset\}$ and $\mathcal{D}_\bot = \emptyset$ which are called `TABLE_DEE` and `TABLE_DUM` (in [39]) and represent the truth values 1 and 0, respectively. Moreover, for each $A \subseteq R$, the restriction of r to the subset A is denoted by $r(A)$, that is $r(A)\colon A \to \bigcup_{y \in A} D_y$. If $\mathcal{D}$ is a data table on a relation scheme R, i.e. $\mathcal{D} \subseteq \mathrm{Tupl}(R)$, and A is a subset of R, then $\pi_A(\mathcal{D})$ denotes the projection of the data table $\mathcal{D}$ to the set of attributes A,

$$\pi_A(\mathcal{D}) = \{r(A) \mid r \in \mathcal{D}\}. \tag{1.19}$$

Assume A, B are sets of attributes, i.e. $A, B \subseteq R$, then we say A determines B (or B is functionally dependent on A) if whenever two tuples of $\mathcal{D}$ agree on attributes from A then they agree on attributes from B. We write $A \Rightarrow B$ and call such a statement functional dependency (FD). Formally, FD is satisfied by relation $\mathcal{D}$ iff

$$\forall r_1, r_2 \in \mathcal{D}:\ \text{if } r_1(A) = r_2(A),\ \text{then } r_1(B) = r_2(B). \tag{1.20}$$

We will denote by $||A \Rightarrow B||_{\mathcal{D}}$ the degree to which an FD $A \Rightarrow B$ holds in a relation $\mathcal{D}$. From (1.20) we obviously have $||A \Rightarrow B||_{\mathcal{D}} \in \{0, 1\}$.

1.2.3 Ranked data tables over domains with similarities

The concept of a ranked data table over domains with similarities [9, 12, 17, 18] is the counterpart to the concept of a relation on a relation scheme. As in the original Codd's relational model, Y denotes a set of attributes names, a relation scheme is any finite subset $R \subseteq Y$, and a domain D_y is a set of all possible values of the attribute $y \in Y$. The relational model is generalized in the following way: i) Each domain D_y is additionally equipped with a similarity relation $\approx_y$, i.e. with a reflexive symmetric binary **L**-relation on D_y; ii) Each tuple has assigned a rank, which represents a degree to which a tuple matches a query. Ranks have mainly comparative meaning: the higher the rank the better the match. Similarity degrees as well as ranks come from complete residuated lattice. The following table which can be seen as a result of the query "hotels in Olomouc with a room for 100 €" is an example of a ranked data table.

	name	*price*	*eval*	*dist*
1.00	Hotel Central	100,00 €	8.9	0.5 km
0.90	Hotel ABC	90,00 €	9.1	0.8 km
0.85	Pension Angel	115,00 €	8.5	1.2 km
0.45	Hotel Paradise	55,00 €	6.7	2.5 km
0.30	Hotel Kryton	170,00 €	10.0	1.6 km

In the data table we store the following informations: ***name*** (name of the hotel), ***price*** (price for the double room), ***eval*** (average evaluation), ***dist*** (distance from the city center). The numbers $1.00, \ldots, 0.30$ in the leftmost column are the ranks from a scale

of truth values (here $[0,1]$). The remaining part of the table can be seen as a classical data table. Similarities on domains are not shown directly. For the attribute *price*, we consider the following similarity on its domain: $p_1 \approx_{price} p_2 = (100 - |p_1 - p_2|)/100$ if $|p_1 - p_2| < 100$, and 0 otherwise.

We will now introduce ranked data tables (RDTs) formally:

Definition 1 ([17]). *Let $R \subseteq Y$ be a relation scheme and let $\langle D_y, \approx_y, \rangle$ be domains with similarities for attributes $y \in R$. A ranked data table on R over $\{\langle D_y, \approx_y, \rangle \,|\, y \in R\}$ is any map*

$$\mathcal{D} : \prod_{y \in R} D_y \to L, \tag{1.21}$$

such that the set $\{r \in \prod_{y\in R} D_y \,|\, \mathcal{D}(r) > 0\}$, called the answer set, is finite. The cardinality of the answer set of $\mathcal{D}$ is called the size *of $\mathcal{D}$ and is denoted by $|\mathcal{D}|$. $\mathcal{D}$ is called* nonranked *if $\mathcal{D}(r) \in \{0,1\}$ for any r. Each degree $\mathcal{D}(r) \in L$ is called a* rank *of r in $\mathcal{D}$.*

Definition 2 ([17]). *For each $a \in L$, we denote by $a_\emptyset$ the RDT on $\emptyset$ such that $a_\emptyset(\emptyset) = a$.*

Therefore, each $a_\emptyset$ is a map which assigns to the empty tuple the degree $a \in L$

$$a_\emptyset \colon \{\emptyset\} \to L. \tag{1.22}$$

Each $a_\emptyset$ is viewed as a relational representation of the rank $a \in L$.

Note that the original Codd's model is a particular case of the model of RDT over domains with similarities. If one takes the two-element Boolean algebra for $\mathbf{L}$, then all RDTs become nonranked and all similarities become identities.

We introduce relational operations for RDTs as given in [17]. For RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$ on relation scheme R, we put

$$(\mathcal{D}_1 \cup \mathcal{D}_2)(r) = \mathcal{D}_1(r) \vee \mathcal{D}_2(r), \tag{1.23}$$

$$(\mathcal{D}_1 \cap \mathcal{D}_2)(r) = \mathcal{D}_1(r) \wedge \mathcal{D}_2(r), \tag{1.24}$$

$$(\mathcal{D}_1 \otimes \mathcal{D}_2)(r) = \mathcal{D}_1(r) \otimes \mathcal{D}_2(r). \tag{1.25}$$

$\mathcal{D}_1 \cup \mathcal{D}_2$ is called the *union* of $\mathcal{D}_1$ and $\mathcal{D}_2$. $\mathcal{D}_1 \cap \mathcal{D}_2$ and $\mathcal{D}_1 \otimes \mathcal{D}_2$ are called the *$\wedge$-intersection* and the *$\otimes$-intersection*.

In order to have a domain independent reisuduum, the authors introduced a ternary counterpart of $\to$ with one of the argument serving as a range [17]. For RDTs $\mathcal{D}_1$, $\mathcal{D}_2$, and $\mathcal{D}_3$ on relational scheme R we put

$$\big(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2\big)(r) = \mathcal{D}_3(r) \otimes \big(\mathcal{D}_1(r) \to \mathcal{D}_2(r)\big) \tag{1.26}$$

for all $r \in \mathrm{Tupl}(R)$. $\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2$ is called a *residuum* of $\mathcal{D}_1$ with respect to $\mathcal{D}_2$ which ranges over $\mathcal{D}_3$. It is clear that $\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2 \subseteq \mathcal{D}_3$. The RDT $\mathcal{D}_3$ serves as a range for the componentwise application of residuum $\to$, which is more easily seen if one considers $\mathcal{D}_3$ as a nonranked RDT. In this case $\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2$ can be rewritten as follows:

$$\big(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2\big)(r) = \begin{cases} \mathcal{D}_1(r) \to \mathcal{D}_2(r), & \text{if } \mathcal{D}(r) = 1, \\ 0, & \text{otherwise.} \end{cases}$$

If we take the first or the second argument of the ternary residuum as a constant degree from L, we obtain two important binary operations: residuated c-negation and residuated c-shift. For RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$ on R and for $c \in L$, we put

$$\big(\mathcal{D}_2 \boxtriangleright_c \mathcal{D}_1\big)(r) = \mathcal{D}_1(r) \twoheadrightarrow^{\mathcal{D}_2(r)} c, \tag{1.27}$$

$$\big(c \twoheadrightarrow^{\mathcal{D}_2} \mathcal{D}_1\big)(r) = c \twoheadrightarrow^{\mathcal{D}_2(r)} \mathcal{D}_1(r), \tag{1.28}$$

for all tuples $r \in \mathrm{Tupl}(R)$.

Projections and residuated divisions represent operations which allow users to express queries with existential and universal quantification.

We start by considering the projection. If $\mathcal{D}$ is an RDT on R_1, the *projection* of $\mathcal{D}$ onto $R_2 \subseteq R_1$, denoted by $\pi_{R_2}(\mathcal{D})$, is defined as

$$(\pi_{R_2}(\mathcal{D}))(r_2) = \bigvee_{r_3 \in \mathrm{Tupl}(R_1 \setminus R_2)} \mathcal{D}(r_2 r_3) \tag{1.29}$$

for each $r_2 \in \mathrm{Tupl}(R_2)$. Note that (1.29) uses a general suprema $\bigvee$ to define the rank of r_2 in $\pi_{R_2}(\mathcal{D})$. If $\mathcal{D}$ is interpreted as a result of query Q, then the rank of r_2 in $\pi_{R_2}(\mathcal{D})$ can be understood as the degree to which "there is a tuple matching Q which agrees with r_2 on all the attributes from R_2".

Relational expressions involving projections can be utilized in existentially quantified queries. In the same spirit, relational expressions involving divisions are algebraic counterpart to universally quantified queries, see [62]. Since in residuated logics the existential and universal quantifiers are not mutually definable [47, 50], the residuated division is introduced as a fundamental operation. Moreover, the residuated division is considered as a ternary operation in order to ensure its domain independence.

Let $\mathcal{D}_1$ be an RDT on R_1, $\mathcal{D}_2$ be an RDT on $R_2 \subseteq R_1$, and $\mathcal{D}_3$ be an RDT on $R_3 = R_1 \setminus R_2$. Then, a *division* of $\mathcal{D}_1$ by $\mathcal{D}_2$ which ranges over $\mathcal{D}_3$ is an RDT on R_3 denoted by $\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2$ and defined by

$$\begin{aligned}\big(\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2\big)(r_3) &= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_2(r_2) \twoheadrightarrow^{\mathcal{D}_3(r_3)} \mathcal{D}_1(r_2 r_3)\big) \\ &= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_3(r_3) \otimes (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2 r_3))\big)\end{aligned} \tag{1.30}$$

for each $r_3 \in \mathrm{Tupl}(R_3)$. It is easily seen that $\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2 \subseteq \mathcal{D}_3$ and therefore the result of division is fully contained in $\mathcal{D}_3$. Therefore $\mathcal{D}_3$ can be seen as a range for the division.

Similarity-based restriction is another fundamental operation and it is a counterpart to the ordinary restriction. If $\mathcal{D}$ is an RDT on relation scheme R, $y \in R$ and $d \in D_y$, the *similarity-based restriction* of $\mathcal{D}$ by $y \approx d$ is an RDT on R denoted by $\sigma_{y \approx d}(\mathcal{D})$ and defined by

$$(\sigma_{y \approx d}(\mathcal{D}))(r) = \mathcal{D}(r) \otimes r(y) \approx_y d, \tag{1.31}$$

for all $r \in \mathrm{Tupl}(R)$. The similarity-based restriction that compares values of two attributes y_1, y_2 with the same domain is introduced as follows: For an RDT $\mathcal{D}$ on relation scheme R and for $y_1, y_2 \in R$ such that $D_{y_1} = D_{y_2}$ and $u \approx_{y_1} v = u \approx_{y_2} v$ for all $u, v \in D_{y_1}$, we define

$$(\sigma_{y_1 \approx y_2}(\mathcal{D}))(r) = \mathcal{D}(r) \otimes r(y_1) \approx_{y_1} r(y_2). \tag{1.32}$$

By applying a similarity-based restriction to a nonranked RDT we obtain a ranked RDT.

The (equality-based) natural join is introduced as follows. If $\mathcal{D}_1$ is an RDT on relation scheme $R_1 \cup R_3$ and $\mathcal{D}_2$ is an RDT of relation scheme $R_2 \cup R_3$ such that $R_1 \cap R_2 = R_1 \cap R_3 = R_2 \cap R_3 = \emptyset$ (i.e., R_1, R_2, and R_3 are pairwise disjoint), then the (*equality-based*) *natural join* of $\mathcal{D}_1$ and $\mathcal{D}_2$ is an RDT on relation scheme $R_1 \cup R_2 \cup R_3$ denoted by $\mathcal{D}_1 \bowtie \mathcal{D}_2$ and defined by

$$\big(\mathcal{D}_1 \bowtie \mathcal{D}_2\big)(r_1 r_2 r_3) = \mathcal{D}_1(r_1 r_3) \otimes \mathcal{D}_2(r_2 r_3) \tag{1.33}$$

for each $r_1 \in \mathrm{Tupl}(R_1)$, $r_2 \in \mathrm{Tupl}(R_2)$, and $r_3 \in \mathrm{Tupl}(R_3)$. Natural joins have important special cases: i) Considering $\mathcal{D}_1$ and $\mathcal{D}_2$ in (1.33) with $R_3 = \emptyset$, we get a natural join $\mathcal{D}_1 \bowtie \mathcal{D}_2$ of two RDTs on disjoint relation schemes; ii) Considering $R_1 = R_2 = \emptyset$, then $\mathcal{D}_1 \bowtie \mathcal{D}_2$ is a $\otimes$-intersection.

Similarity-based restrictions can be used to define various types of similarity-based joins. The first type of join we introduce is a similarity-based equijoin. For RDTs $\mathcal{D}_1$ on R_1 and $\mathcal{D}_2$ on R_2 such that $R_1 \cap R_2 = \emptyset$, the *similarity-based equijoin* of $\mathcal{D}_1$ and $\mathcal{D}_2$ by $y_1 \approx y_2$, denoted by $\mathcal{D}_1 \bowtie_{y_1 \approx y_2} \mathcal{D}_2$, is defined by

$$\mathcal{D}_1 \bowtie_{y_1 \approx y_2} \mathcal{D}_2 = \sigma_{y_1 \approx y_2}(\mathcal{D}_1 \bowtie \mathcal{D}_2), \tag{1.34}$$

provided that $y_1 \in R_1$, $y_2 \in R_2$, and both y_1 and y_2 have the same domain with similarity. A second type of join can be used when we want to put only a partial emphasis instead of the full emphasis on the similarity-based condition $y_1 \approx y_2$:

$$(\mathcal{D}_1 \bowtie_{c/y_1 \approx y_2} \mathcal{D}_2)(r_1 r_2) = \mathcal{D}_1(r_1) \otimes \mathcal{D}_2(r_2) \otimes (c \to r_1(y_1) \approx_{y_1} r_2(y_2)) \tag{1.35}$$

for any $r_1 \in \mathrm{Tupl}(R_1)$ and $r_2 \in \mathrm{Tupl}(R_2)$.

We have seen that similarity-based restriction can produce a ranked RDT from a nonranked one. Conversely, operations kernel and support produce a nonranked RDT from a data table containing ranks. For $\mathcal{D}$, we define RDTs $\Delta\mathcal{D}$ (a *kernel* of $\mathcal{D}$) and $\nabla\mathcal{D}$ (a *support* of $\mathcal{D}$) on the same relation scheme as follows:

$$(\Delta\mathcal{D})(r) = \begin{cases} 1, & \text{if } \mathcal{D}(r) = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{1.36}$$

$$(\nabla\mathcal{D})(r) = \begin{cases} 1, & \text{if } \mathcal{D}(r) > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{1.37}$$

The last operation we discuss is renaming, which plays the same role as in the Codd's model. Given an RDT $\mathcal{D}$, the *renaming* $\rho_f(\mathcal{D})$ produces an RDT with the same contents (and the same ranks) with attributes renamed by an injective renaming function $f : R \to Y$ such that attributes y and $f(y)$ have the same domain.

The authors proved in [14, 17] that relational algebra has the same expressive power as the domain relational calculus (with range declarations). The domain relational calculus is based on first-order fuzzy logic.

Similarity-based functional dependencies

We now introduce functional dependencies, which are called similarity-based functional dependencies (SBFDs) and their interpretation in RDTs [9, 12, 15, 18]. For $A, B \in L^R$

the similarity-based functional dependency is an expression of the form $A \Rightarrow B$. For an RDT $\mathcal{D}$ on R a degree $||A \Rightarrow B||_{\mathcal{D}}$ to which $A \Rightarrow B$ is true in $\mathcal{D}$ is defined by

$$||A \Rightarrow B||_{\mathcal{D}} = \bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} \Big((r_1(A) \approx r_2(A))^* \rightarrow (r_1(B) \approx r_2(B)) \Big), \tag{1.38}$$

where

$$r_1(A) \approx_{\mathcal{D}} r_2(A) = (\mathcal{D}(r_1) \otimes \mathcal{D}(r_2)) \rightarrow \bigwedge_{y \in R} (A(y) \rightarrow r_1(y) \approx_y r_2(y)). \tag{1.39}$$

In what follows, we are interested in the entailment of SBFDs from theories [9, 18]. An $\mathbf{L}$-set T of SBFDs on R will be called a *theory.* A theory T is called *crisp* if $T(A \Rightarrow B) \in \{0, 1\}$ for each SBFD $A \Rightarrow B$. We say that an RDT $\mathcal{D}$ is a model of theory T whenever $T(A \Rightarrow B) \leq ||A \Rightarrow B||_{\mathcal{D}}$ for all $A \Rightarrow B$ on R. The collection of models will be denoted as $\mathrm{Mod}(T)$, i.e.

$$\mathrm{Mod}(T) = \{\mathcal{D} \,|\, \text{for each } A, B \in L^R : T(A \Rightarrow B) \leq ||A \Rightarrow B||_{\mathcal{D}}\}, \tag{1.40}$$

where $\mathcal{D}$ is any RDT over R. A degree $||A \Rightarrow B||_T$ to which $A \Rightarrow B$ (on R) *semantically follows* from T is defined by

$$||A \Rightarrow B||_T = \textstyle\bigwedge_{\mathcal{D} \in \mathrm{Mod}(T)} ||A \Rightarrow B||_{\mathcal{D}}. \tag{1.41}$$

The degree to which a particular SBFD follows from a given theory (an $\mathbf{L}$-set of SBFDs) can be expressed using the concepts of entailment to degree 1 and crisp theory. More precisely: For $A, B \in L^R$ and theory T on R

$$||A \Rightarrow B||_T = \bigvee \{c \in L \,|\, ||A \Rightarrow c \otimes B||_{\mathrm{crisp}(T)} = 1\}, \tag{1.42}$$

where $\mathrm{crisp}(T) = \{A \Rightarrow T(A \Rightarrow B) \otimes B \,|\, A, B \in L^R \text{ and } T(A \Rightarrow B) \otimes B \not\subseteq A\}$.

We have introduced the concept of semantic entailment, which is defined in terms of models. As in the ordinary Codd model there is an alternative type of entailment based on the notion of provability. The deductive system for SBFDs consists of three rules:

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

(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 $A, B, C, D \in L^R$ and $c \in L$. The inference system consisting of (Ax), (Cut), and (Mul) is complete in the following sense: $||A \Rightarrow B||_T = 1$ iff $T \vdash A \Rightarrow B$, i.e., iff there is a proof of $A \Rightarrow B$ from T. The proof of $A \Rightarrow B$ from T is a sequence of SBFDs ending with $A \Rightarrow B$ such that each element of the sequence is either from T or is inferred from the preceding formulas using (Ax), (Mul), or (Cut). This result (ordinary-style completeness) characterizes SBFDs which follow semantically from T to degree 1. There is also a result on graded-style (Pavelka-style [68, 69, 70]) completeness saying that

$$||A \Rightarrow B||_T = \bigvee \{c \in L \,|\, T \vdash A \Rightarrow c \otimes B\}, \tag{1.43}$$

i.e., the degree to which $A \Rightarrow B$ semantically follows from T is a supremum of degrees $c \in L$ for which $A \Rightarrow c \otimes B$ is provable from T in the ordinary sense. The completeness results have been established over all finite residuated lattices and general complete residuated lattices (considering an additional infinitary deduction rule), see [8].

Chapter 2

Overview of similarity-based functional dependencies

We have seen one particular extension of functional dependencies, namely similarity-based functional dependencies proposed by Belohlavek and Vychodil. This approach is one of many extensions that appeared in the past, actually more than one hundred papers dealing with functional dependencies (FDs) over domain with similarities can be found in the literature. In our opinion the wide variety of approaches are worthy of an exhaustive review and comparison.

The name "fuzzy functional dependencies" is often used for various extensions of FDs which we think is unfortunate for several reasons. First of all, the term "fuzzy functional dependency" is usually used for functional dependencies defined within "fuzzy relational model". But there is no agreement among researchers what the terms "fuzzy relational model" or "fuzzy database" really mean, compare for example [84] and [73]. Moreover, although many definitions of so called fuzzy functional dependencies extend the classical one, the dependency usually remains crisp in the sense that either a given relation satisfies the dependency or it does not. In this sense the term fuzzy functional dependency is somehow inadequate. We will therefore use the term generalized functional dependency (GFD) and generalized relational model (GRM) to prevent misunderstanding.

In this chapter, we intend to concentrate specifically on GFD over domains with similarities and the directly related issues. From the logical point of view, the generalization of FDs to FDs over domains with similarities may be looked at as replacing two-valued identity relations by many-valued ones which represent similarities. This step may be considered as switching from a two-valued logic, as the formal framework in which the ordinary model is implicitly developed [1], to appropriate fuzzy logic. The switch to fuzzy logic naturally brings the question of how the concept of validity, entailment etc. should be dealt with. Should the validity of a GFD in a given relation remains bivalent (true or false) or should the validity be many-valued (e.g. taking values from $[0, 1]$)?

There are several works which addressed and examined the various proposals to GFDs, [21,

[1] Codd's original model was based on two-valued logic, although later Codd himself extended its relational calculus by considering a three-valued logic to manage missing, non-applicable or unknown information via the NULL value [31]. In the further step [32] a four-valued logic was introduced to deal separately with these different types of uncertainties. Nevertheless, these extensions have been subject to criticism in the past (see C.J. Date in [37]).

22, 15, 88]. None of the works is trying to unify various approaches or to objectively compare them.

2.1 Generalizations of the relational model

The main goal of this section is to look into generalizations of the relational model involving similarity relations. The extension of domains with a similarity relation usually does not stand alone but comes together with ranked data tables, and various data extensions.

1. **Similarity-based approaches (from equality to similarity)**: In most of the approaches we will consider, the equality relation that is implicitly presented in the original Codd's model (domain values are either "equal" or "not equal") is replaced by a binary fuzzy relation that maps every pair of domain values to $[0, 1]$ and is meant to express the similarity (closeness) of domain values [20, 23, 24, 35, 26, 41, 71, 74, 75, 76, 91, 80] and later in [9, 19, 44, 56, 60, 78, 83, 95]. Although the degree of similarity comes usually from $[0, 1]$, there are extensions considering more general algebraic structures [49, 9, 33].

2. **Rank-based approaches (from relation to fuzzy relation)**: By rank-based approaches we mean extensions of the relational model in which the data table is seen as a fuzzy set of tuples. Thus the data table has an additional column which contains a rank—also called (membership) grade, score or weight—to express to what degree a tuple belongs to a data table. First attempts to rank-based approaches can be found in [2, 53, 74, 85, 96]. Later works include [15, 90, 43, 64, 66, 75, 81, 83]. There are also extensions in which the rank is assigned to every attribute value, e.g. in [63, 33, 16]. The ranks usually take values from $[0, 1]$, but there are approaches in which the unit interval is replaced by some general algebraic structure [48, 49, 15]. In one of the pioneering work done by Umano [85] the rank itself is a possibility distribution on $[0, 1]$.

   The meaning of the rank differs among approaches and it is seen as: (i) compatibility with the relation [2]; (ii) global confidence level [22]; (iii) compatibility with the set of individual constraints specified on the relation, see [64, 66]; (iv) degree to which a tuple matches a query, see [15, 43, 75], or the degree to which it is possible that a tuple matches a query, see [25].

3. **Data extensions (from precise to imprecise values)**:
   The third aspect involved in the various generalizations of the relational model is data extensions, i.e. replacing precise values by imprecise ones. There are several approaches where the authors are trying to incorporate more complex data, namely an attribute value is considered to be a set of (possible) values in [24, 26, 79, 91, 89], a fuzzy set (including linquistic terms) or a possibility distribution in [20, 67, 36, 41, 26, 55, 57, 60, 74, 66, 72, 81, 85], a vague set in [95] or an interval-valued possibility distribution in [65]. Nevertheless, in Codd's relational model there are no limitations in what can and cannot be an attribute value [38].

2.2 Comparison of similarity-based generalizations of FD

The semantics of classical FD corresponds to the notion of mathematical function. More precisely: $||A \Rightarrow B||_{\mathcal{D}} = 1$ iff $\{\langle r(A), r(B)\rangle | \forall r \in \mathcal{D}\}$ is a function (from $\pi_A(\mathcal{D})$ to $\pi_B(\mathcal{D})$, see (1.19)). In this section we will introduce various definitions of GFD and examine how different approaches correspond to the notion of a fuzzy function. The definition of fuzzy function was provided by Gottwald in [46] and later studied for example by Demirci [40].

Definition 3 (Fuzzy function)**.** *Let* $\mathbf{L}$ *be a residuated lattice, let* A *and* B *be crisp sets, and let* $\approx_A$ *and* $\approx_B$ *be* $\mathbf{L}$*-equalities. An* $\mathbf{L}$*-relation* $\rho\colon A \times B \to L$ *(*L *is a support set of* $\mathbf{L}$*) is said to be a fuzzy function iff for all* $a_1, a_2 \in A$ *and* $b_1, b_2 \in B$ *we have*

$$\rho(a_1, b_1) \otimes \rho(a_2, b_2) \otimes (a_1 \approx_A a_2) \leq (b_1 \approx_B b_2). \tag{2.1}$$

A partial fuzzy function [40, 54] is used in [4] in the definition of a degree to which a given relation is a fuzzy function. We will use the idea from [46] (and later from [4]) to define a degree to which a ranked data table corresponds to the notion of fuzzy function given by (2.1).

Definition 4. *Let* $\mathbf{L}$ *be a complete residuated lattice and* $\mathcal{D} : \mathrm{Tupl}(R) \to L$ *be a ranked data table. Let* $\approx_i$ *be* $\mathbf{L}$*-similarities on corresponding domains. Let* $A, B \subseteq R$ *and let the similarity of two tuples on a set of attributes be given by Equation* (2.6)*. The degree to which* $\mathcal{D}$ *is a fuzzy function with respect to the sets of attributes* A *and* B *is defined as:*

$$\mathrm{Fun}(\mathcal{D}, A, B) = \bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} \Big((\mathcal{D}(r_1) \otimes \mathcal{D}(r_2) \otimes (r_1(A) \approx_{\mathcal{D}} r_2(A))) \to (r_1(B) \approx_{\mathcal{D}} r_2(B)) \Big). \tag{2.2}$$

Definition 4 gives us the degree to which a relation (data table $\mathcal{D}$) captures the notion of fuzzy function from A to B. The following criterion will give us the degree to which: "For all relations $\mathcal{D}$: If a GFD $A \Rightarrow B$ is satisfied in relation $\mathcal{D}$, then $\mathcal{D}$ corresponds to the fuzzy function from A to B."

$$\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \big(||A \Rightarrow B||_{\mathcal{D}} \to \mathrm{Fun}(\mathcal{D}, A, B) \big). \tag{2.3}$$

Similarly, the next criterion will give us a degree to which: "For all relations $\mathcal{D}$: If $\mathcal{D}$ corresponds to the fuzzy function from A to B, then a GFD $A \Rightarrow B$ is satisfied by $\mathcal{D}$."

$$\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \big(\mathrm{Fun}(\mathcal{D}, A, B) \to ||A \Rightarrow B||_{\mathcal{D}} \big). \tag{2.4}$$

Finally, combining (2.3) and (2.4) we will obtain the degree to which a particular definition of GFD corresponds to the fuzzy function,

$$\begin{aligned} \mathbb{E}(\mathrm{Fun}, A \Rightarrow B) &= \bigwedge_{\mathcal{D}:\mathrm{Tupl}(R)\to L} \big(\mathrm{Fun}(\mathcal{D}, A, B) \leftrightarrow ||A \Rightarrow B||_{\mathcal{D}} \big) \\ &= \mathbb{S}(A \Rightarrow B, \mathrm{Fun}) \wedge \mathbb{S}(\mathrm{Fun}, A \Rightarrow B). \end{aligned} \tag{2.5}$$

Survey of similarity-based generalizations of FD

If not otherwise stated we assume a relation scheme $R = \{y_1, \ldots, y_n\}$, $A, B \subseteq R$. Similarity or equivalence relation on domain D_i of attribute y_i will be denoted as $\approx_i$, $\equiv_i$, respectively. We assume that the similarity of two tuples r_1, r_2 from some data table $\mathcal{D}$ on R is defined as:

$$r_1(A) \approx_{\mathcal{D}} r_2(A) = \min_{y_i \in A} r_1(y_i) \approx_i r_2(y_i). \tag{2.6}$$

If there is no confusion we will write $r_1(A) \approx r_2(A)$ instead of $r_1(A) \approx_{\mathcal{D}} r_2(A)$.

Buckles and Petry (1983): One of the pioneering work was done by Buckles and Petry, see [24]. The authors introduced a model, in which domains are equipped with fuzzy equivalence relations (called similarity in the original work) and tuple values are allowed to be (ordinary) non-empty subsets of the domain.

Generalized functional dependencies (called fuzzy FD) were defined in [1]. Later, the definition was modified and reformulated using the so called conformance [91, 92]: Let $0 < \beta \leq 1$. The GFD $A \Rightarrow_\beta B$ holds in the Buckles-Petry model, iff for every pair of tuples r_i, r_j:

$$\beta * (r_1(A) \approx r_2(A)) \leq (r_1(B) \approx r_2(B)), \tag{2.7}$$

where $*$ is the arithmetic product.

Theorem 5. *Let* $\mathbf{L} = [0,1]_{\Pi}$*, let* $A, B \subseteq R$ *be sets of attributes and let the GFD be defined as in* (2.7)*. Assuming* $\beta \in [0,1]$ *is the parameter from* (2.7)*, then* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = \beta$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \beta \to 0$*.*

Prade and Testemale(1984): In [72] Prade and Testemale considered so called possibilistic fuzzy data model, i.e. attribute values are allowed to be possibility distribution in Zadeh's sense [94]. A model based on the concept of possibility distribution was originally proposed by Umano [85]. The relation $\mathcal{D}$ is defined as:

$$\mathcal{D} \subseteq \prod_{y \in R} [0,1]^{D_y \cup \{e\}}, \tag{2.8}$$

where $[0,1]^{D_y \cup \{e\}}$ denotes the set of all possibility distributions on $D_y \cup \{e\}$. Moreover, each domain $D_y \cup \{e\}$ is associated with a similarity relation (called fuzzy proximity relation) $\sim_y$ which takes values from $[0,1]$. The similarity relation is then extended to possibility distributions on $D_y \cup \{e\}$.

The GFDs were introduced only for singleton sets. Given a fixed threshold $\lambda \in [0,1]$ and $y_i, y_j \in R$, the GFD $\{y_i\} \Rightarrow \{y_j\}$ is satisfied in $\mathcal{D}$ if and only if for all $r_1, r_2 \in \mathcal{D}$

$$(r_1(y_i) = r_2(y_i)) \to (r_1(y_j) \approx_j r_2(y_j) \geq \lambda), \tag{2.9}$$

where $\to$ is the ordinary implication.

Theorem 6. *Let* $\mathbf{L}$ *be any complete residuated lattice with universe* $L = [0,1]$*. Let* $A, B \in R$*. Let the GFD be defined by* (2.9) *and let* $\lambda \in [0,1]$ *be the parameter from* (2.9)*. Then* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 0$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \lambda \to 0$*.*

Raju and Majumdar (1988): Another generalization of FD was proposed by Raju and Majumdar [74]. They considered similarity relation on each domain and ranks associated to each tuple, but the ranks and similarity degrees come from [0.1]. More precisely, a relation $\mathcal{D}$ is a fuzzy subset on Tupl(R):

$$\mathcal{D} : \prod_{y \in R} D_y \to [0,1]. \tag{2.10}$$

Therefore every tuple r has associated a degree (rank) to which the tuple belongs to $\mathcal{D}$, denoted as $\mathcal{D}(r)$. The meaning of the ranks in not clearly given. The generalized functional dependency $A \Rightarrow B$ is satisfied by a relation $\mathcal{D}$ iff for all $r_1, r_2 \in \mathcal{D}$ ($r_1, r_2 \in \mathcal{D}$ means $r_1, r_2 \in \text{Tupl}(R)$ with $\mathcal{D}(r_1) > 0$ and $\mathcal{D}(r_2) > 0$)

$$r_1(A) \approx r_2(A) \leq r_1(B) \approx r_2(B). \tag{2.11}$$

Theorem 7. *Let* $\mathbf{L}$ *be any complete residuated lattice with universe* $L = [0,1]$*. Assume* R *is a relational scheme and* $A, B \subseteq R$*. For the GFD given by Equation* (2.11)*,* $\mathbb{S}(A \Rightarrow B, \text{Fun}) = 1$ *and* $\mathbb{S}(\text{Fun}, A \Rightarrow B) = 0$*.*

The paper by Raju and Majumdar is probably the most influential one and the definition of the GFD given by (2.11) inspired many authors [58, 57, 59, 55, 60, 77, 95].

Chen (1991): Another significant proposal of definition of GFD was developed by Chen [28], see also [27, 26]. Chen used the possibilistic fuzzy data model:

$$\mathcal{D} \subseteq \prod_{y \in R} [0,1]^{D_y}, \tag{2.12}$$

where $[0,1]^{D_y}$ denotes the set of all possibility distributions over domain D_y. Moreover, a similarity relation $\sim_y$ (originally called closeness relation) is associated with each domain D_y, which is then used to express the similarity $\approx_y$ of attribute values (possibility distribution). The GFD $A \Rightarrow B$ holds in $\mathcal{D}$ to a degree θ iff for all pair of tuples r_1, r_2:

$$\begin{aligned} &\text{if } r_1(A) = r_2(A) \text{ then } r_1(B) = r_2(B), \\ &(r_1(A) \approx r_2(A) \to_G r_1(B) \approx r_2(B)) \geq \theta \text{ otherwise.} \end{aligned} \tag{2.13}$$

The fact that $||A \Rightarrow B||_{\mathcal{D}} = \theta$ does not exclude existence of other $\theta' > \theta$ for which the inequality (2.13) holds.

Theorem 8 (Chen et al.)**.** *Let* $\mathbf{L} = [0,1]_G$*,* $A, B \subseteq R$ *be sets of attributes,* $\theta \in [0,1]$ *and let the GFD be defined as in* (2.13)*. Then* $\mathbb{S}(A \Rightarrow B, \text{Fun}) = 1$ *and* $\mathbb{S}(\text{Fun}, A \Rightarrow B) = 0$*.*

Bhuniya and Niyogi (1993): According to Bhuniya and Niyogi [20] the generalized functional dependency $A \Rightarrow B$ holds in a Raju-Majumdar's model (2.10) if and only if for all $r_1, r_2 \in \mathcal{D}$ one of the following conditions holds

$$\begin{aligned} &r_1(A) \approx r_2(A) \leq r_1(B) \approx r_2(B), \\ &r_1(A) \approx r_2(A) - r_1(B) \approx r_2(B) \leq 1 - \beta, \end{aligned} \tag{2.14}$$

where $r_1(A) \approx r_2(A) \geq \alpha$, $r_1(B) \approx r_2(B) \geq \alpha$, and $\alpha < \beta < 1$.

Theorem 9. *Let* $\mathbf{L} = [0,1]_L$, $A, B \subseteq R$. *For GFD given by* (2.14) *we have* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = \beta$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 0$.

Cubero et al. (1994): Cubero et al. [36] proposed the following definition of an GFD for a possibilistic fuzzy data model (2.12). Each domain D_y is equipped with similarity relation (called proximity in the original work) and a fixed threshold c_y. GFD $A \Rightarrow B$ is satisfied iff for all $r_1, r_2 \in \mathcal{D}$:

$$(r_1(A) \approx r_2(A) \geq \alpha) \rightarrow (r_1(B) \approx r_2(B) \geq \beta). \tag{2.15}$$

The parameters α and β are vectors, $\alpha = (c_y)_{y \in A}$, $\beta = (c_y)_{y \in B}$, where values $c_y \in [0,1], y \in R$, are fixed and common to all GFDs.

Theorem 10. *Let* $\mathbf{L}$ *be any complete residuated lattice with universe* $L = [0,1]$. *For the GFD given by Equation* (2.15) *and for fixed thresholds* c_y, $y \in R$:

$$\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = \big(\bigvee_{y \in A} c_y \rightarrow 0 \big) \wedge \bigwedge_{y \in B} c_y, \tag{2.16}$$

$$\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = \big(\bigwedge_{y \in A} c_y \rightarrow \bigvee_{y \in B} c_y \big) \rightarrow 0. \tag{2.17}$$

Ben Yahia et al. (1999): The authors considered the Raju-Majumdar's model with uncertain data (fuzzy sets) and ranks coming from $[0,1]$. The GFD is defined as follows: A determines B to degree β, denoted as $A \sim\!>_\beta B$, $\beta, \theta \in [0,1]$ in $\mathcal{D}$ if for all tuples r_1 and r_2 we have:

$$(r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B)) \geq \theta, \tag{2.18}$$

where

$$\beta = \min_{r_1, r_2} (r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B)), \tag{2.19}$$

and $\rightarrow$ is the Łukasiewicz implication. The threshold θ is fixed by the database designer.

Theorem 11. *Let* R *be a relational scheme and* $A, B \subseteq R$. *For GFD given by* (2.18) *we have* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 1$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 0$.

Bosc et al. (1999): Another generalization was done by Bosc, Pivert, and Ughetto, see [23]. They were the first that used residuated implication corresponding to some t-norm. A GFD is defined as:

$$\forall r_1, r_2 \in \mathcal{D} : r_1(A) \approx r_2(A) \rightarrow r_1(B) \approx r_2(B). \tag{2.20}$$

Unfortunately, the authors presented only those definitions and did not go any further by showing properties of such FD or presenting inference rules. Nevertheless, our criterion (2.5) is satisfied to degree 1.

Tyagi et al. (2005): Later Tyagi et al. [83] introduced another generalization of functional dependencies using the framework of fuzzy functions (3) The authors developed GFD for Raju and Majumdar's model. Relation $\mathcal{D}$ satisfies the GFD $A \Rightarrow B$ if its projection over $A \cup B$ (denoted as $\mathcal{D}_{AB}$) is a partial fuzzy function. That is, if $\forall r_1, r_2 \in \mathrm{Tupl}(A \cup B)$:

$$(\mathcal{D}_{AB}(r_1) \wedge \mathcal{D}_{AB}(r_2) \wedge r_1(A) \equiv r_2(A)) \leq r_1(B) \equiv r_2(B), \tag{2.21}$$

where $\mathcal{D}_{AB}(r) = \sup\{\mathcal{D}(r') \mid r' \in \mathrm{Tupl(R)} \text{ such that } r'(A \cup B) = r\}$.

Theorem 12 (Tyagi et al.)**.** *Let* $\mathbf{L} = [0,1]_G$*,* $A, B \subseteq R$ *be sets of attributes and let the GFD be defined as in* (2.21)*. Then* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 1$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 0$*.*

Kiss (1991): The idea that the rank should influence the validity of FD can be already found in [53]. The truth value to which the fuzzy relation $\mathcal{D}$ satisfies a given FD was given by

$$||A \Rightarrow B||_{\mathcal{D}} = 1 - \sup\{\inf(\mathcal{D}(r_1), \mathcal{D}(r_2)) |\, r_1(A) = r_2(A) \text{ and } r_1(B) \neq r_2(B))\}. \tag{2.22}$$

It is clear from (2.22) that the higher the degree of $\mathcal{D}(r_1)$ and $\mathcal{D}(r_2)$ when r_1, r_2 violate the classical FD, the lower the the truth degree of FD $A \Rightarrow B$.

Theorem 13. *Let* $\mathbf{L} = [0,1]_L$*,* $A, B \subseteq R$ *be sets of attributes and let the GFD be defined as in* (2.22)*. Then* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 1$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 0.5$*.*

Cordero et al. (2011): The last extension we want to mention in this section was presented by Cordero et al. in [34]. The authors worked with a generalization of Codd's relational model called fuzzy attribute table. The basic idea is that tuple value has assigned a rank coming from complete residuated lattice. More precisely, the fuzzy attribute table is understood as a map

$$\mathcal{D}\colon \prod_{y \in R} D_y \to L^R. \tag{2.23}$$

This means that for each tuple r: $\mathcal{D}(r) \in L^R$, i.e. $\mathcal{D}(r)$ is a tuple of truth values. For all $y \in R$, $\mathcal{D}(r)(y)$ is the truthfulness of tuple r in the value $r(y)$.

The authors introduced the following definition: a fuzzy attribute table $\mathcal{D}$ is said to satisfy a generalized functional dependency $A \Rightarrow B$ with θ degree iff

$$\theta \leq \bigwedge_{r_1, r_2 \in \mathrm{Tupl}(R)} (r_1(A) \approx_{\mathcal{D}} r_2(A)) \to (r_1(B) \approx_{\mathcal{D}} r_2(B)), \tag{2.24}$$

where $\to$ is a residuated implication. In [34] the authors considered supremum of degrees to which the GFD is true. That is:

$$||A \Rightarrow B||_{\mathcal{D}} = \sup\{\theta \in [0,1] \mid \theta \text{ satisfies } (2.24)\}.$$

Theorem 14 (Cordero et al. case)**.** *Let* R *be a relational scheme and* $A, B \subseteq R$*. If the GFD is defined by Equation* (2.24)*, then* $\mathbb{S}(A \Rightarrow B, \mathrm{Fun}) = 1$ *and* $\mathbb{S}(\mathrm{Fun}, A \Rightarrow B) = 1$*.*

The summary of results established in this chapter can be found in Table 2.1. We want to emphasize several points:

1. As it is shown in Table 2.1, many approaches reduce the new (generalized) concept of FD to a bivalent one.

2. In some cases the interpretation of a rank is not very clear.

3. The ranks are not usually involved in the definition of GFD. This fact yields to odd behavior: tuples with very low ranks may caused the GFD to be satisfied to low degree (even 0).

Authors/approach	GFD	*[Imp]*	*[TrGFD]*	*[Rank]*	$\mathbb{E}(\text{Fun}, A \Rightarrow B)$
Buckles and Petry [24]	(2.7)	R-G imp.	$\{0, 1\}$	No	$\beta \wedge (\beta \to 0)$
Prade and Testemale [72]	(2.9)	R-G imp.	$\{0, 1\}$	No	0
Raju and Majumdar [74]	(2.11)	R-G imp.	$\{0, 1\}$	Yes	0
Chen et al. [45]	(2.13)	Classical or Gödel	$\{0, 1\}$	No	0
Bhuniya and Niyogi [20]	(2.14)	R-G imp.	$\{0, 1\}$	Yes	0
Cubero et al. [36]	(2.15)	R-G imp.	$\{0, 1\}$	No	(2.16)$\wedge$(2.17)
Tyagi et al. [83]	(2.21)	R-G imp.	$\{0, 1\}$	Yes	0
Kiss [53]	(2.22)	Łukasiewicz	$[0, 1]$	Yes	0.5
Ben Yahia et al. [90]	(2.18)	Łukasiewicz	$[0] \cup [\theta, 1]$	Yes	0
Bosc, Pivert and Ughetto [23]	(2.20)	Residuum	$[0, 1]$	No	1
Belohavek and Vychodil [15]	(2.18)	Residuum	Complete residuated lattice	Yes	1
Cordero et al. [34]	(2.24)	Residuum	$[0, 1]$	Yes	1

Table 2.1: Review of similarity-based functional dependencies.
In the *[Imp]* column the implication used in definition of a GFD is highlighted. The choice of the implication influences the degree to which a GFD is true, column *[TrGFD]*. The column *[Rank]* indicates if a GFD is defined for data table with ranks. In the last column the degree to which a GFD corresponds to fuzzy function is presented.

None of these problems appears in approaches which are built on fuzzy logic in narrow sense [15, 34]. Among conceptual clarity, the connection to fuzzy logic in narrow sense enables us to generalize many concepts form the original Codd's relational model (which is connected to the first order logic). In the next chapter we will provide a generalization of derivation graphs [61] which can be seen as an alternative prove system.

Chapter 3

Derivation digraphs for graded if-then rules

In this chapter we present a graph-based method of reasoning with graded if-then rules, by which we mean rules of the form $A \Rightarrow B$, where A, B are fuzzy sets of attributes. Rules of this form describe dependencies between attributes in ordinal and similarity-based data and have two basic interpretations: 1) Similarity-based functional dependencies, see Section 1.2.3, which are interpreted in ranked data tables; 2) Attribute implications (AIs) in formal concept analysis with grades [7]. The notion of semantic entailment for SBFDs coincide with the notion of semantic entailment for AIs in FCA with grades in the following sense: The degree to which a graded if-then rule $A \Rightarrow B$ follows from a theory (**L**-set of graded if-then rules) is the same under both interpretations [7]. As a consequence, on may use single Armstrong-like axiomatization, for example the rules (Ax), (Cut) and (Mul) from Section 1.2.3.

Looking for a graph-based inference system for graded if-then rules is interesting from several viewpoints. First, the notion of semantic entailment of the rules we consider is graded, i.e., the entailment expresses a degree to which a rule follows from other rules. It is therefore interesting to find a graph-based inference system that is able to infer rules from other ones including the entailment degrees. Second, there is an Armstrong-like axiomatization of the semantic entailment for the graded rules (see Section 1.2.3, or the original papers [9, 7]), i.e., one might be interested in finding a corresponding graph-based inference method. Third, the Armstrong-like proofs can be formalized to form particular sequences (so-called MRAP-sequences, see [11]). It is therefore interesting to observe whether the graph-based proofs can be constructed according to the normalized proofs and *vice versa.*

In what follows the graded if-then rules will be called fuzzy attribute implications (FAIs). We will first introduce derivation digraphs as particular labeled acyclic digraphs constructed from an input theory (collections of FAIs).

3.1 Derivation acyclic digraphs for FAIs

We now introduce derivation digraphs as particular acyclic digraphs where vertices are labeled by attributes from R and degrees from **L**. The arcs of the digraphs will correspond

to FAIs from an input theory and indicate which formulas from the theory are used in the process of inference. In what follows, $\mathbf{L}$ is a complete residuated lattice. In order to denote that * is a hedge on $\mathbf{L}$, we write $\mathbf{L}^*$.

Definition 15 (T-based $\mathbf{L}^*$-derivation DAG)**.** Let T be a set of FAIs over R.

1. Any $\mathbf{D} = \langle V, \emptyset \rangle$ such that $\emptyset \neq V \subseteq R \times L$ and for every $y \in R$ there is at most one $a \in L$ such that $\langle y, a \rangle \in V$, is a T-based $\mathbf{L}^*$-derivation DAG;
2. If $\mathbf{D} = \langle V, A \rangle$ is a T-based $\mathbf{L}^*$-derivation DAG and there are $E \Rightarrow F \in T$, attribute $y \in R$, and vertices $\langle y_1, a_1 \rangle \in V, \ldots, \langle y_k, a_k \rangle \in V$ such that for

$$s_0 = \bigwedge \{E(y) \to 0 \,|\, y \in R \text{ and } y \notin \{y_1, \ldots, y_k\}\}, \tag{3.1}$$
$$s_1 = \bigwedge \{E(y_i) \to a_i \,|\, i = 1, \ldots, k\}, \tag{3.2}$$
$$m = \bigvee \{a \in L \,|\, \langle y, a \rangle \in V\}, \tag{3.3}$$
$$d = \big((s_0 \wedge s_1)^* \otimes F(y)\big) \vee m, \tag{3.4}$$

we have $d > m$, then $\mathbf{D}' = \langle V', A' \rangle$, where

$$V' = V \cup \{\langle y, d \rangle\}, \tag{3.5}$$
$$A' = A \cup \{\langle \langle y_i, a_i \rangle, \langle y, d \rangle \rangle \,|\, i = 1, \ldots, k\}, \tag{3.6}$$

is a T-based $\mathbf{L}^*$-derivation DAG.

If $\mathbf{D}$ is a T-based $\mathbf{L}^*$-derivation DAG, we put

$$\mathbf{D}(y) = \bigvee \{a \in L \,|\, \langle y, a \rangle \in V\}, \tag{3.7}$$

and call $\mathbf{D}(y)$ the *yield of* $\mathbf{D}$ *on* y. Clearly, the yield of $\mathbf{D}$ corresponds to (3.3), i.e., we can interpret it as the degree to which y is assumed to be valid according to $\mathbf{D}$. Moreover, $\langle y, a \rangle \in V$ is called an *initial vertex* of $\mathbf{D}$ if $\langle y, a \rangle$ has no incoming arcs (i.e., no arc in $\mathbf{D}$ enters $\langle y, a \rangle$).

Notice that for each $y \in R$ such that $\mathbf{D}(y) > 0$ there is $\langle y, a \rangle \in V$ such that $a = \mathbf{D}(y)$. This is a consequence of Definition 15. Furthermore, it follows that for any $y \in R$, the set

$$L_y = \{a \in L \,|\, \langle y, a \rangle \in V\} \tag{3.8}$$

has a greatest element provided that $L_y \neq \emptyset$. Another direct consequence of Definition 15 is that L_y is either empty or it is a finite subchain (if equipped with the restriction of $\leq$ to L_y) of the lattice part of $\mathbf{L}$. The latter observation is of course trivial if $\mathbf{L}$ is a chain but it pertains to all complete residuated lattices taken for $\mathbf{L}$. We make use of these observations later in the proofs.

The following notion introduces derivation digraphs related to FAIs:

Definition 16 (T-based $\mathbf{L}^*$-derivation DAG for $E \Rightarrow F$)**.** Let $\mathbf{D} = \langle V, A \rangle$ be a T-based $\mathbf{L}^*$-derivation DAG. Then $\mathbf{D}$ is called a T-based $\mathbf{L}^*$-derivation DAG for $E \Rightarrow F$ if the following conditions are all satisfied:

1. $\mathbf{D}(y) \geq F(y)$ for all $y \in R$;

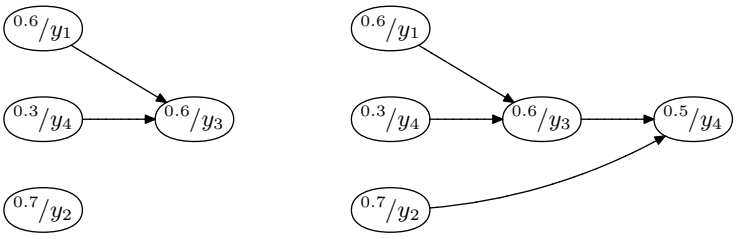


Figure 3.1: Construction of T-based $\mathbf{L}^*$-derivation DAG.

2. if $E \neq \emptyset$ then the set of initial vertices of $\mathbf{D}$ is

$$\{\langle y, E(y)\rangle \mid y \in R \text{ and } E(y) > 0\}; \tag{3.9}$$

3. if $E = \emptyset$, then the set of initial vertices of $\mathbf{D}$ is $\{\langle y^\sharp, 0\rangle\}$,

where $y^\sharp \in R$ is a designated attribute.

By the designated attribute in the previous definition we mean a fixed attribute that has been selected from R (no particular role or intended interpretation of the attribute is assumed).

Example 1. *In this example, we utilize the residuated lattice with $L = [0, 1]$ given by the Łukasiewicz operations together with hedge * defined as follows: For each $a \in L$ we put*

$$a^* = \begin{cases} 1, & \text{for } a = 1, \\ 0.6, & \text{for } 0.6 \leq a \leq 0.9, \\ 0.2, & \text{for } 0.2 \leq a \leq 0.5, \\ 0, & \text{for } 0 \leq a \leq 0.1. \end{cases}$$

Figure 3.1 depicts a single step of the process of construction of a T-based $\mathbf{L}^*$-derivation DAG for FAI $\{^{0.6}/y_1, ^{0.7}/y_2, ^{0.3}/y_4\} \Rightarrow \{^{0.5}/y_4\}$, where T is the following set of FAIs:

$$\begin{aligned} T = \{&\{^{0.7}/y_1, ^{0.5}/y_4\} \Rightarrow \{^{0.7}/y_2, ^{1}/y_3\}, \{^{0.7}/y_3\} \Rightarrow \{^{0.8}/y_5\}, \\ &\{^{0.7}/y_2, ^{0.9}/y_3\} \Rightarrow \{^{0.9}/y_4\}\}. \end{aligned}$$

The DAG on the right-hand side of Figure 3.1 results from the DAG on the left-hand side by adding vertex $\langle y_4, 0.5\rangle$ and two arcs leading from $\langle y_2, 0.7\rangle$ and $\langle y_3, 0.6\rangle$.

3.2 Completeness

We now turn our attention to the completeness by which we mean a characterization of the semantic entailment by existence of $\mathbf{L}^*$-derivation DAGs. We prove the claim by showing that a FAI is provable from a theory T iff it has a T-based $\mathbf{L}^*$-derivation DAG. We now show that T-based $\mathbf{L}^*$-derivation DAGs are in a correspondence with normalized proofs called MRAP-sequences [11].

Recall from [11] that the following three rules can be derived from (Ax), (Cut) and (Mul), which were introduced in Section 1.2.3:

(Ref) infer $A \Rightarrow A$,

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

(Pro) from $A \Rightarrow B \cup C$ infer $A \Rightarrow B$,

for all $A, B, C, D, E \in L^Y$. The rules are called reflexivity, accumulation and projection, respectively. By a derivable rule we mean that for all $A, B, C, D, E \in L^Y$, from the part preceding "infer", we can derive using (Ax), (Mul), and (Cut), the part succeeding "infer".

By an *MRAP-sequence for $A \Rightarrow B$ from T* (see [11]), we mean a sequence of formulas such that it

(a) starts with $A \Rightarrow A$;

(b) continues with FAIs from T;

(c) continues with FAIs which result from using (Mul) on FAIs from (b);

(d) continues with FAIs which result from using (Acc) on FAIs from (a), (b), (c), (d);

(e) ends with a single application of (Pro), on the last FAI in (d);

(f) the FAI which results by (e) is $A \Rightarrow B$.

Theorem 17. *Let T be a theory. If there is an MRAP-sequence for $A \Rightarrow B$ from T, then there is a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$.*

In the opposite direction, we have the following characterization.

Theorem 18. *Let T be a theory. If there is a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$, then there is an MRAP-sequence for $A \Rightarrow B$ from T.*

The following assertion provides the ordinary-style completeness:

Theorem 19. *If $\mathbf{L}$ is finite, then $||A \Rightarrow B||_T = 1$ iff there is a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$.*

Furthermore, we can express the graded-style completeness as follows:

Theorem 20. *If $\mathbf{L}$ is finite, then $||A \Rightarrow B||_T$ is the greatest degree $a \in L$ such that there is a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow a \otimes B$.*

In abstract fuzzy logic (also known as Pavelka's fuzzy logic [50, 68, 69, 70]), theories as considered as $\mathbf{L}$-sets of formulas. $T(\varphi)$ is interpreted as a degree to which T prescribes φ valid. Even in this case, we can show that T-based $\mathbf{L}^*$-derivation DAGs are capable of describing degrees of semantic entailment as it is shown by the following theorem.

Theorem 21. *If $\mathbf{L}$ is finite and T is an $\mathbf{L}$-set of FAIs, then $||A \Rightarrow B||_T$ is the greatest degree $a \in L$ such that there is a T'-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow a \otimes B$, where $T' = \{A \Rightarrow T(A \Rightarrow B) \otimes B \mid A, B \in L^Y$ and $T(A \Rightarrow B) \otimes B \not\subseteq A\}$.*

3.3 Computing closures

Considering the construction of T-based $\mathbf{L}^*$-derivation DAGs as an alternative proof technique not only can help visualize the inference from if-then rules but in addition, the construction of such DAGs yields algorithms for checking whether (and to what degree) $A \Rightarrow B$ semantically follows from a theory. Indeed, in order to check whether $||A \Rightarrow B||_T = 1$, we may proceed as follows:

Procedure 22 (Checking of full entailment)**.** *For any T and $A \Rightarrow B$:*

1. *Construct a T-based $\mathbf{L}^*$-derivation DAG $\mathbf{D} = \langle V, \emptyset \rangle$ with*

$$V = \{\langle y, A(y) \rangle \mid A(y) > 0\};$$

   If $V = \emptyset$, put $V = \{\langle y^\sharp, 0 \rangle\}$, where $y^\sharp$ is the designated attribute, see Definition 16.

2. *If $\mathbf{D}(y) \geq B(y)$ for all $y \in R$, stop and return "YES"; otherwise continue with step 3.*

3. *If $\mathbf{D}$ can be enlarged according to Definition 15 (case 2.), then enlarge $\mathbf{D}$ and continue with step 2.; otherwise return "NO".*

Theorem 23. *Assuming $\mathbf{L}$ finite, for any $A \Rightarrow B$ and theory T, Procedure 22 terminates after finitely many steps and it returns "YES" iff $||A \Rightarrow B||_T = 1$.*

The previous observations enable us to extend the procedure for computing syntactic closures for $\mathbf{L}$-sets of attributes. Let us recall that by a *closure of A under T*, denoted A_T^+, we mean the largest $\mathbf{L}$-set such that $T \vdash A \Rightarrow A_T^+$, see [7]. For every A and T, A_T^+ always exists, is uniquely given and has the following important property [7] provided that $\mathbf{L}$ is finite:

$$||A \Rightarrow B||_T = S\big(B, A_T^+\big). \tag{3.10}$$

The closure A_T^+ can be obtained from the yield of a final T-based $\mathbf{L}^*$-derivation DAG:

Definition 24 (Final T-based $\mathbf{L}^*$-derivation DAG)**.** A T-based $\mathbf{L}^*$-derivation DAG is called final if there are no $E \Rightarrow F \in T$, attribute $y \in R$, and vertices $\langle y_1, a_1 \rangle \in V, \ldots, \langle y_k, a_k \rangle \in V$ such that for s_0, s_1, m, d given by (3.1)–(3.4) we have $d > m$.

Theorem 25. *Let $\mathbf{L}$ be finite and $\mathbf{D}$ be a T-based $\mathbf{L}^*$-derivation DAG for $A \Rightarrow B$. Then $\mathbf{D}$ is final iff $\mathbf{D}(y) = A_T^+(y)$ for all $y \in R$.*

Owing to Theorem 25, in order to compute the degree $||A \Rightarrow B||_T$, it suffices to find a single final T-based $\mathbf{L}^*$-derivation DAG $\mathbf{D}$ for $A \Rightarrow B$, and apply (3.10) for A_T^+ determined from the yield of $\mathbf{D}$. We may formalize this computation by a modification of Procedure 22:

Chapter 4

Sensitivity analysis for similarity-based functional dependencies

In this chapter we look at similarity estimates for SBFDs (given by Equation (1.38)) in ranked data tables (RDTs) over domains with similarities (see Section 1.2.3). We answer some natural questions such as: What is the relationship between $||A \Rightarrow B||_{\mathcal{D}_1}$ and $||A \Rightarrow B||_{\mathcal{D}_2}$ in terms of similarity of RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$? Or what can we say about the truth degrees $||A \Rightarrow B_1||_{\mathcal{D}}$ and $||A \Rightarrow B_2||_{\mathcal{D}}$ in terms of similarity of B_1 and B_2. The first problem we discuss in this chapter is how to assess similarity of two ranked data tables.

4.1 Rank-based similarity

In this section, we introduce a notion of a similarity and a related notion of a graded containment (subsethood) of RDTs on the same relation scheme R. As in the case of domain similarities, the similarity of RDTs is expressed by degrees from the complete residuated lattice $\mathbf{L}$.

The rank-based similarity of RDTs which is based on the idea that RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$ (on the same relation scheme R) are similar iff for each tuple $r \in \mathrm{Tupl}(R)$, ranks $\mathcal{D}_1(r)$ and $\mathcal{D}_2(r)$ are similar degrees from $\mathbf{L}$. Similarity of degrees from $\mathbf{L}$ can be expressed by a biresiduum (1.2). Since we are interested in assessing similarity of $\mathcal{D}_1(r)$ and $\mathcal{D}_2(r)$ for all possible tuples r, we may define the *similarity* $E(\mathcal{D}_1, \mathcal{D}_2)$ of RDTs $\mathcal{D}_1$ and $\mathcal{D}_2$ as an infimum which goes over all tuples:

$$E(\mathcal{D}_1, \mathcal{D}_2) = \bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \leftrightarrow \mathcal{D}_2(r)\big). \tag{4.1}$$

An alternative (but equivalent) way to define similarity of RDTs is the following:

$$S(\mathcal{D}_1, \mathcal{D}_2) = \bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \rightarrow \mathcal{D}_2(r)\big), \tag{4.2}$$

$$E(\mathcal{D}_1, \mathcal{D}_2) = S(\mathcal{D}_1, \mathcal{D}_2) \wedge S(\mathcal{D}_2, \mathcal{D}_1). \tag{4.3}$$

4.2 Similarity estimates for FD

We have seen how to define the similarity of two RDTs on the same relation scheme. An interesting question regarding the validity of SBFDs is: What can we say about the truth degree of $A \Rightarrow B$ in similar RDTs? Recall from (1.3) that $a^2 = a \otimes a$.

Theorem 26. *For RDTs $\mathcal{D}_1$, $\mathcal{D}_2$ on the same relation scheme R we have*

$$(S(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes S(\mathcal{D}_2, \mathcal{D}_1)^2 \leq ||A \Rightarrow B||_{\mathcal{D}_1} \rightarrow ||A \Rightarrow B||_{\mathcal{D}_2}, \tag{4.4}$$

$$(E(\mathcal{D}_1, \mathcal{D}_2)^*)^2 \otimes E(\mathcal{D}_2, \mathcal{D}_1)^2 \leq ||A \Rightarrow B||_{\mathcal{D}_1} \leftrightarrow ||A \Rightarrow B||_{\mathcal{D}_2}. \tag{4.5}$$

If we take an identity for a hedge, the left hand side of the Equation (4.5) can be simplified.

Corollary 27. *For RDTs $\mathcal{D}_1$, $\mathcal{D}_2$ on the same relation scheme R and for hedge being identity we have*

$$E(\mathcal{D}_1, \mathcal{D}_2)^4 \leq ||A \Rightarrow B||_{\mathcal{D}_1} \leftrightarrow ||A \Rightarrow B||_{\mathcal{D}_2}.$$

We now turn our attention to hedges. Since a hedge is used as a parameter in the definition of SBFD, the natural question is how the truth degree of SBFD $A \Rightarrow B$ when hedge $*_1$ is used differs from the truth degree of the same SBFD when hedge $*_2$ is used. In order to emphasize the hedge used in the definition of SBFD, we will employ the following notation: $||A \Rightarrow B||^*_{\mathcal{D}}$. First of all, we need to capture the similarity of two hedges:

Definition 28 ([10])**.** *For hedges $*_1, *_2$ on $\mathbf{L}$ put*

$$(*_1 \preceq *_2) = \bigwedge_{a \in L} (a^{*_1} \rightarrow a^{*_2}), \tag{4.6}$$

$$(*_1 \approx *_2) = \bigwedge_{a \in L} (a^{*_1} \leftrightarrow a^{*_2}). \tag{4.7}$$

The Equation (4.7) can be interpreted as a degree to which hedges $*_1$ and $*_2$ yield similar results. More precisely, (4.7) is a true degree of of the following formula: "for each $a \in L$: the result of a^{*_1} is similar to the result of a^{*_2}." Analogously, (4.6) can be interpreted as a degree to which $*_1$ is stronger than $*_2$.

Theorem (29) shows that "if $A \Rightarrow B$ is true using hedge $*_2$ and if hedge $*_1$ is stronger than $*_2$, then $A \Rightarrow B$ is true using hedge $*_1$" and that "if the hedges $*_1$ and $*_2$ are similar, then the degrees to which $A \Rightarrow B$ is true using hedge $*_1$ and hedge $*_2$ are similar".

Theorem 29. *Let $A, B \in \mathbf{L}^R$ and let $*_1$, $*_2$ be two hedges on $\mathbf{L}$. Then for any RDT $\mathcal{D}$ on R we have:*

$$(*_1 \preceq *_2) \leq ||A \Rightarrow B||^{*_2}_{\mathcal{D}} \rightarrow ||A \Rightarrow B||^{*_1}_{\mathcal{D}}, \tag{4.8}$$

$$(*_1 \approx *_2) \leq ||A \Rightarrow B||^{*_2}_{\mathcal{D}} \leftrightarrow ||A \Rightarrow B||^{*_1}_{\mathcal{D}}. \tag{4.9}$$

The next problem we want to tackle is how the truth degree of $A \Rightarrow B$ depends on the truth degrees prescribed by the $\mathbf{L}$-sets A, B.

Lemma 30. *Let $A, B_1, B_2 \in \mathbf{L}^R$ be fuzzy sets of attributes. For any RDT $\mathcal{D}$ on R we have*

$$S(B_2, B_1) \otimes ||A \Rightarrow B_1||_{\mathcal{D}} \leq ||A \Rightarrow B_2||_{\mathcal{D}}. \tag{4.10}$$

Lemma 31. *Let $A_1, A_2, B \in \mathbf{L}^R$ be fuzzy sets of attributes. For any RDT $\mathcal{D}$ on R we have*

$$S(A_1, A_2)^* \otimes ||A_1 \Rightarrow B||_{\mathcal{D}} \leq ||A_2 \Rightarrow B||_{\mathcal{D}}. \tag{4.11}$$

To sum up, we obtain the following Theorem.

Theorem 32. *Let $A_1, A_2, B_1, B_2 \in \mathbf{L}^R$. For any RDT $\mathcal{D}$ on R and for fixed hedge $*$ we have:*

$$S(A_1, A_2)^* \otimes S(B_2, B_1) \otimes ||A_1 \Rightarrow B_1||_{\mathcal{D}} \leq ||A_2 \Rightarrow B_2||_{\mathcal{D}}, \tag{4.12}$$

$$E(A_1, A_2)^* \otimes E(B_2, B_1) \leq ||A_1 \Rightarrow B_1||_{\mathcal{D}} \leftrightarrow ||A_2 \Rightarrow B_2||_{\mathcal{D}}. \tag{4.13}$$

Chapter 5

Similarity estimates of query results

In this chapter we will show that relational operations from Section 1.2.3 are robust because they are *insensitive to slight changes in data*: (very) similar input data cannot yield (very) different results under the notions of similarity defined by (4.1). This has many practical implications. For instance, if two experts are asked to assign ranks in a datatable based on their knowledge of particular problem domain, they can come up with different ranks. If the assigned ranks are sufficiently close, we know that we can take either of the ranked data tables and it will produce similar results as the other one when used in subsequent queries. Later in this chapter we will provide an alternative measure of similarity of RDTs based on ranks and tuple values, and we will introduce related relational operation—a similarity-based closure.

5.1 Similarity estimates for relational operations

We describe the similarity estimates for relational operations from Section 1.2.3.

1) Boolean-like operation:

The following assertion shows that $\cup$ and $\cap$ preserve subsethood degrees and similarity degrees given by (4.2) and (4.1), respectively.

Theorem 33. *For any $\mathcal{D}_1$, $\mathcal{D}'_1$, $\mathcal{D}_2$, and $\mathcal{D}'_2$ on relation scheme R,*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2), \tag{5.1}$$

$$S(\mathcal{D}_1, \mathcal{D}'_1) \wedge S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \cap \mathcal{D}_2, \mathcal{D}'_1 \cap \mathcal{D}'_2), \tag{5.2}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \wedge E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2), \tag{5.3}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \wedge E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \cap \mathcal{D}_2, \mathcal{D}'_1 \cap \mathcal{D}'_2). \tag{5.4}$$

The estimates we will investigate further in this section employ $\otimes$ instead of $\wedge$ for combining subsethood degrees. Since $\wedge$ is an upper bound for $\otimes$ in $\mathbf{L}$ we have

$$S(\mathcal{D}_1, \mathcal{D}_2) \otimes S(\mathcal{D}'_1, \mathcal{D}'_2) \leq S(\mathcal{D}_1, \mathcal{D}_2) \wedge S(\mathcal{D}'_1, \mathcal{D}'_2)$$

and analogously for E. Therefore the Corollary 34 immediately follows from Theorem 33.

Corollary 34. *For any $\mathcal{D}_1$, $\mathcal{D}'_1$, $\mathcal{D}_2$, and $\mathcal{D}'_2$ on relation scheme R,*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2), \tag{5.5}$$
$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \cap \mathcal{D}_2, \mathcal{D}'_1 \cap \mathcal{D}'_2), \tag{5.6}$$
$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}'_1 \cup \mathcal{D}'_2), \tag{5.7}$$
$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \cap \mathcal{D}_2, \mathcal{D}'_1 \cap \mathcal{D}'_2). \tag{5.8}$$

Let us note that inclusion estimates like those from Theorem 33 do not have a nontrivial interpretation in the original Codd's model of data.

2) Ternary residuum:

Theorem 35. *For any $\mathcal{D}_1$, $\mathcal{D}'_1$, $\mathcal{D}_2$, $\mathcal{D}'_2$, $\mathcal{D}_3$, and $\mathcal{D}'_3$ on R, we have:*

$$S(\mathcal{D}'_1, \mathcal{D}_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \leq S(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2, \mathcal{D}'_1 \twoheadrightarrow^{\mathcal{D}'_3} \mathcal{D}'_2), \tag{5.9}$$
$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \otimes E(\mathcal{D}_3, \mathcal{D}'_3) \leq E(\mathcal{D}_1 \twoheadrightarrow^{\mathcal{D}_3} \mathcal{D}_2, \mathcal{D}'_1 \twoheadrightarrow^{\mathcal{D}'_3} \mathcal{D}'_2). \tag{5.10}$$

Corollary 36. *For any $\mathcal{D}_1, \mathcal{D}'_1, \mathcal{D}_2, \mathcal{D}'_2$ on R:*

$$S(\mathcal{D}'_1, \mathcal{D}_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c \to c') \leq S(\mathcal{D}_2 \boxtriangleright_c \mathcal{D}_1, \mathcal{D}'_2 \boxtriangleright_{c'} \mathcal{D}'_1), \tag{5.11}$$
$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c \leftrightarrow c') \leq E(\mathcal{D}_2 \boxtriangleright_c \mathcal{D}_1, \mathcal{D}'_2 \boxtriangleright_{c'} \mathcal{D}'_1), \tag{5.12}$$
$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c' \to c) \leq S(c \twoheadrightarrow^{\mathcal{D}_2} \mathcal{D}_1, c' \twoheadrightarrow^{\mathcal{D}'_2} \mathcal{D}'_1), \tag{5.13}$$
$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c \leftrightarrow c') \leq E(c \twoheadrightarrow^{\mathcal{D}_2} \mathcal{D}_1, c' \twoheadrightarrow^{\mathcal{D}'_2} \mathcal{D}'_1). \tag{5.14}$$

3) Projection and division:

Theorem 37. *Let $\mathcal{D}$ and $\mathcal{D}'$ be RDTs on relation scheme R_1 and let $R_2 \subseteq R_1$. Then*

$$S(\mathcal{D}, \mathcal{D}') \leq S(\pi_{R_2}(\mathcal{D}), \pi_{R_2}(\mathcal{D}')), \tag{5.15}$$
$$E(\mathcal{D}, \mathcal{D}') \leq E(\pi_{R_2}(\mathcal{D}), \pi_{R_2}(\mathcal{D}')). \tag{5.16}$$

Now we turn our attention to residuated division (1.30), which was also introduced in the Section 1.2.3. First, let us note that residuated division can be used to express containment and similarity degrees of RDTs. Consider the borderline case of residuated division when $R_1 = R_2$ (and thus $R_3 = \emptyset$) and $\mathcal{D}_3 = 1_\emptyset$:

$$\begin{aligned}(\mathcal{D}_1 \div^1 \mathcal{D}_2)(\emptyset) &= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} \big(1 \otimes (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2))\big) \\ &= \textstyle\bigwedge_{r_2 \in \mathrm{Tupl}(R_2)} (\mathcal{D}_2(r_2) \to \mathcal{D}_1(r_2)) = S(\mathcal{D}_2, \mathcal{D}_1).\end{aligned}$$

As a consequence, (4.2) and (4.1) are expressible inside the model of Belohlavek and Vychodil and thus the similarity estimations are relational *per se*.

The similarity estimates for residuated division are described by the following theorem.

Theorem 38. *Let $\mathcal{D}_1$, $\mathcal{D}'_1$ be RDTs on R_1, $\mathcal{D}_2$, $\mathcal{D}'_2$ be RDTs on $R_2 \subseteq R_1$, and $\mathcal{D}_3$, $\mathcal{D}'_3$ be RDTs on $R_3 = R_1 \setminus R_2$, respectively. Then*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}'_2, \mathcal{D}_2) \otimes S(\mathcal{D}_3, \mathcal{D}'_3) \leq S(\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2, \mathcal{D}'_1 \div^{\mathcal{D}'_3} \mathcal{D}'_2), \tag{5.17}$$
$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \otimes E(\mathcal{D}_3, \mathcal{D}'_3) \leq E(\mathcal{D}_1 \div^{\mathcal{D}_3} \mathcal{D}_2, \mathcal{D}'_1 \div^{\mathcal{D}'_3} \mathcal{D}'_2). \tag{5.18}$$

4) Similarity-based restriction:

Theorem 39. *Let $\mathcal{D}$ and $\mathcal{D}'$ be RDTs on relation scheme R and let $y \in R$ and $d \in D_y$. Then,*

$$S(\mathcal{D}, \mathcal{D}') \leq S(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx d}(\mathcal{D}')), \tag{5.19}$$

$$E(\mathcal{D}, \mathcal{D}') \leq E(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx d}(\mathcal{D}')). \tag{5.20}$$

The similarity estimates in Theorem 39 involve two restrictions using the same constant d from the domain of y. Intuitively, we may expect that two restrictions that use different constants d and d' should yield similar results if d and d' are similar. This can be shown if the similarity on the domain of y is $\otimes$-transitive.

Theorem 40. *Let $\mathcal{D}$ be an RDT on R, let $y \in R$, $d, d' \in D_y$, and let $\approx_y$ be $\otimes$-transitive. Then*

$$d \approx_y d' \leq S(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx d'}(\mathcal{D})). \tag{5.21}$$

Corollary 41. *Let $\mathcal{D}$ and $\mathcal{D}'$ be RDTs on R and let $y \in R$, $d, d' \in D_y$ and $\approx_y$ be $\otimes$-transitive. Then,*

$$S(\mathcal{D}, \mathcal{D}') \otimes d \approx_y d' \leq S(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx d'}(\mathcal{D}')), \tag{5.22}$$

$$E(\mathcal{D}, \mathcal{D}') \otimes d \approx_y d' \leq E(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx d'}(\mathcal{D}')). \tag{5.23}$$

Further question related to similarity is whether a small change of the definition of domain similarities yields a small change of query results. This type of similarity preservation can also be established in the model.

In order to measure containment and similarity of domain similarities, we introduce the following notation. If $\approx_y$ and $\approx'_y$ are similarities on the same domain D_y, we put:

$$S(\approx_y, \approx'_y) = \textstyle\bigwedge_{d_1, d_2 \in D_y} \big(d_1 \approx_y d_2 \rightarrow d_1 \approx'_y d_2\big), \tag{5.24}$$

$$E(\approx_y, \approx'_y) = \textstyle\bigwedge_{d_1, d_2 \in D_y} \big(d_1 \approx_y d_2 \leftrightarrow d_1 \approx'_y d_2\big). \tag{5.25}$$

Theorem 42. *Let $\mathcal{D}$ be RDT on R, $y \in R$, and $d \in D_y$. Furthermore, let $\approx_y$ and $\approx'_y$ be similarities on D_y. Then*

$$S(\mathcal{D}, \mathcal{D}') \otimes S(\approx_y, \approx'_y) \leq S(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx' d}(\mathcal{D}')), \tag{5.26}$$

$$E(\mathcal{D}, \mathcal{D}') \otimes E(\approx_y, \approx'_y) \leq E(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx' d}(\mathcal{D}')). \tag{5.27}$$

Corollary 43. *Let $\mathcal{D}$ and $\mathcal{D}'$ be RDTs on R, let $y \in R$, and let $\approx_y$ and $\approx'_y$ be similarities on D_y. Then,*

$$S(\approx_y, \approx'_y) \leq S(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx' d}(\mathcal{D})), \tag{5.28}$$

$$E(\approx_y, \approx'_y) \leq E(\sigma_{y \approx d}(\mathcal{D}), \sigma_{y \approx' d}(\mathcal{D})). \tag{5.29}$$

5) Natural and similarity-based joins:

We will first explore the similarity preservation for the equality-based natural join, and utilize observations on similarity preservation of other operations to get estimates for other joins.

Theorem 44. *Let $\mathcal{D}_1$, $\mathcal{D}'_1$ be RDTs on $R_1 \cup R_3$ and $\mathcal{D}_2$, $\mathcal{D}'_2$ be RDTs on $R_2 \cup R_3$ such that $R_1 \cap R_2 = R_1 \cap R_3 = R_2 \cap R_3 = \emptyset$. Then*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \bowtie \mathcal{D}_2, \mathcal{D}'_1 \bowtie \mathcal{D}'_2), \tag{5.30}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \bowtie \mathcal{D}_2, \mathcal{D}'_1 \bowtie \mathcal{D}'_2). \tag{5.31}$$

Corollary 45. *For any $\mathcal{D}_1$, $\mathcal{D}'_1$, $\mathcal{D}_2$, and $\mathcal{D}'_2$ on relation scheme R,*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \otimes \mathcal{D}_2, \mathcal{D}'_1 \otimes \mathcal{D}'_2), \tag{5.32}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \otimes \mathcal{D}_2, \mathcal{D}'_1 \otimes \mathcal{D}'_2). \tag{5.33}$$

Corollary 46. *For any $\mathcal{D}, \mathcal{D}'$ on R:*

$$S(\mathcal{D}, \mathcal{D}') \otimes (c \to c') \leq S(c \otimes \mathcal{D}, c' \otimes \mathcal{D}'), \tag{5.34}$$

$$E(\mathcal{D}, \mathcal{D}') \otimes (c \leftrightarrow c') \leq E(c \otimes \mathcal{D}, c' \otimes \mathcal{D}'). \tag{5.35}$$

Corollary 47. *Let $\mathcal{D}_1$ and $\mathcal{D}'_1$ be RDTs on R_1 and let $\mathcal{D}_2$ and $\mathcal{D}'_2$ be RDTs on R_2 such that $R_1 \cap R_2 = \emptyset$. Let $y_1 \in R_1$ and $y_2 \in R_2$ have the same domain with similarity. Then*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \leq S(\mathcal{D}_1 \bowtie_{y_1 \approx y_2} \mathcal{D}_2, \mathcal{D}'_1 \bowtie_{\bowtie_{y_1 \approx y_2}} \mathcal{D}'_2), \tag{5.36}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \leq E(\mathcal{D}_1 \bowtie_{y_1 \approx y_2} \mathcal{D}_2, \mathcal{D}'_1 \bowtie_{y_1 \approx y_2} \mathcal{D}'_2). \tag{5.37}$$

Theorem 48. *Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be RDTs on R_1 and R_2 such that $R_1 \cap R_2 = \emptyset$. Let $y_1 \in R_1$ and $y_2 \in R_2$ have the same domain with similarity and let $c, c' \in L$. Then*

$$c' \to c \leq S\big(\mathcal{D}_1 \bowtie_{c/_{y_1 \approx y_2}} \mathcal{D}_2, \mathcal{D}_1 \bowtie_{c'/_{y_1 \approx y_2}} \mathcal{D}_2\big), \tag{5.38}$$

$$c \leftrightarrow c' \leq E\big(\mathcal{D}_1 \bowtie_{c/_{y_1 \approx y_2}} \mathcal{D}_2, \mathcal{D}_1 \bowtie_{c'/_{y_1 \approx y_2}} \mathcal{D}_2\big). \tag{5.39}$$

Corollary 49. *Let $\mathcal{D}_1$ and $\mathcal{D}'_1$ be RDTs on R_1 and let $\mathcal{D}_2$ and $\mathcal{D}'_2$ be RDTs on R_2 such that $R_1 \cap R_2 = \emptyset$. Let $y_1 \in R_1$ and $y_2 \in R_2$ have the same domain with similarity. Then,*

$$S(\mathcal{D}_1, \mathcal{D}'_1) \otimes S(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c' \to c) \leq S(\mathcal{D}_1 \bowtie_{c/_{y_1 \approx y_2}} \mathcal{D}_2, \mathcal{D}'_1 \bowtie_{c'/_{y_1 \approx y_2}} \mathcal{D}'_2), \tag{5.40}$$

$$E(\mathcal{D}_1, \mathcal{D}'_1) \otimes E(\mathcal{D}_2, \mathcal{D}'_2) \otimes (c' \leftrightarrow c) \leq E(\mathcal{D}_1 \bowtie_{c/_{y_1 \approx y_2}} \mathcal{D}_2, \mathcal{D}'_1 \bowtie_{c'/_{y_1 \approx y_2}} \mathcal{D}'_2), \tag{5.41}$$

for all $c, c' \in L$.

6) Further operations:

So far, we have shown that relational operations introduced in Section 1.2.3 preserve similarities and as a consequence, pairwise similar arguments to operations yield similar results. Among the operations we have not considered yet are the operations of renaming, kernel, and support which also belong to the basic operations in the model.

Since the operation of renaming only changes the names of attributes without altering the data table as such (i.e. data as well as ranks stay untouched), the renaming preserves similarity trivially. As we have seen in Section 1.2.3, kernel and support are unary operations that produce a nonranked table from an RDT. It is easily seen that by nature, neither the kernel nor the support preserve similarity except for the trivial cases. The negative result on preserving similarity by kernel and support should not be interpreted as a weakness of the model. For the majority of queries which are free of kernels and supports, one can utilize all the positive results we have made in this section. In practice, the kernel and supports are used as the "outermost operations", so one can always estimate similarity of the results prior to the application of kernels and supports.

5.2 Similarity of complex query results

In this section, we extend the previous results from single operations to arbitrarily complex relational queries.

First, we formalize relational algebra expressions which constitute queries [14, 17]. We assume a fixed *database scheme* which is given by a finite set of relation symbols $\mathbb{r}_1, \ldots, \mathbb{r}_n$, each relation symbol $\mathbb{r}_i$ is given its relation scheme. Furthermore, we assume that all attributes appearing in the schemes of relation symbols have defined their domains. In this setting, the *relational algebra expressions* (shortly, *RA-expressions*) are defined as follows:

1. If $\mathbb{r}$ is a relation symbol on scheme R, then $\mathbb{r}$ is RA-expression on scheme R;
2. if $a \in L$, then $a_\emptyset$ is RA-expression on $\emptyset$;
3. if Q_1 and Q_2 are RA-expressions on R, then $(Q_1 \cap Q_2)$ and $(Q_1 \cup Q_2)$ are RA-expressions on R;
4. if Q_1, Q_2, and Q_3 are RA-expressions on R, then $(Q_1 \twoheadrightarrow^{Q_3} Q_2)$ is RA-expression on R;
5. if Q_1 is RA-expression on R_1 and Q_2 is RA-expression on R_2 then $(Q_1 \bowtie Q_2)$ is RA-expression on $R_1 \cup R_2$; if $R_1 \cap R_2 = \emptyset$, $c \in L$, $y_1 \in R_1$, $y_2 \in R_2$, and both attributes y_1 and y_2 have the same domain, then $(Q_1 \bowtie_{y_1 \approx y_2} Q_2)$ and $(Q_1 \bowtie_{c/y_1 \approx y_2} Q_2)$ are RA-expressions on $R_1 \cup R_2$;
6. if Q is RA-expression on R_1 and $R_2 \subseteq R_1$, then $\pi_{R_2}(Q)$ is RA-expression on R_2;
7. if Q_1 is RA-expression on R_1, Q_2 is RA-expression on $R_2 \subseteq R_1$, and Q_3 is RA-expression on $R_3 = R_1 \setminus R_2$, then $(Q_1 \div^{Q_3} Q_2)$ is RA-expression on R_3;
8. if Q is RA-expression on R, $y \in R$, and $d \in D_y$ (d is a value from the domain of y), then $\sigma_{y \approx d}(Q)$ is RA-expression on R; if $z \in R$ has the same domain as y, then $\sigma_{y \approx z}(Q)$ is RA-expression on R;
9. if Q is RA-expression on R, and f is an injective map such that $f(y)$ has the same domain as y ($y \in R$), then $\rho_f(Q)$ is RA-expression on $h(R)$.

In addition, if Q is RA-expression on R, we call R the *relation scheme of* Q.

As usual, we may evaluate RA-expressions in databases instances to get results of queries. In our setting, a *database instance* $\boldsymbol{\mathcal{D}}$ consists of RDTs which interpret the relation symbols and defines similarities on domains. In a more detail, for each relation variable $\mathbb{r}_i$ from the database scheme, a database instance $\boldsymbol{\mathcal{D}}$ defines its interpretation denoted $\mathbb{r}_i^{\boldsymbol{\mathcal{D}}}$ (an RDT) so that the relation scheme of $\mathbb{r}_i$ is the same as the scheme of $\mathbb{r}_i^{\boldsymbol{\mathcal{D}}}$. Moreover, for each attribute y, $\boldsymbol{\mathcal{D}}$ defines the similarity $\approx_y^{\boldsymbol{\mathcal{D}}}$ on its domain. The notion of database instance is presented here in a simplified form but it is sufficient for the subsequent considerations.

Given an RA-expression Q on scheme R and a database instance $\boldsymbol{\mathcal{D}}$, we denote by $Q^{\boldsymbol{\mathcal{D}}}$ the value of Q in $\boldsymbol{\mathcal{D}}$ which is an RDT on scheme R defined recursively by cases (as usual). Now, we may ask the following question:

Do similar queries yield similar results when evaluated in similar database instances?

By a similar query, we mean a query which results from other query by modifying some of its subqueries. For instance, if a query Q_1 involves a similarity-based restriction using constant d, we may consider its modification Q_2 by substituting d' for d and preserving the rest of this query. Then, considering two database instances $\boldsymbol{\mathcal{D}}_1$ and $\boldsymbol{\mathcal{D}}_2$, we may be interested in estimating the similarity degree $E\big(Q_1^{\boldsymbol{\mathcal{D}}_1}, Q_2^{\boldsymbol{\mathcal{D}}_2}\big)$, i.e., the degree to which $Q_1^{\boldsymbol{\mathcal{D}}_1}$ (the result of Q_1 in $\boldsymbol{\mathcal{D}}_1$) and $Q_2^{\boldsymbol{\mathcal{D}}_2}$ (the result of Q_2 in $\boldsymbol{\mathcal{D}}_2$) are similar.

In order to formalize the similarity estimates, for a pair of queries Q_1 and Q_2, we define their similarity $E(Q_1, Q_2)$ as a map of the form

$$E(Q_1, Q_2)\colon \mathcal{I} \times \mathcal{I} \to L, \tag{5.42}$$

where $\mathcal{I}$ is a set of all database instances of the considered database scheme. Thus, for database instances $\boldsymbol{\mathcal{D}}_1$ and $\boldsymbol{\mathcal{D}}_2$, $\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2)$ is a degree from L. Our intention is to define the degree so that it is a lower bound of the similarity of $Q_1^{\boldsymbol{\mathcal{D}}_1}$ and $Q_2^{\boldsymbol{\mathcal{D}}_2}$.

We define (5.42) by cases taking into account the structure of Q_1 and Q_2. In the following list, we use $\stackrel{\text{def}}{=}$ to denote that the left-hand size of assignment expressions with $\stackrel{\text{def}}{=}$ is defined whenever the right-hand side is defined. Following the definition of RA-expressions, we distinguish the following cases:

- If Q_1 and Q_2 are relation symbols $\mathbb{r}_1$ and $\mathbb{r}_2$ on the same relation scheme, then

$$\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} E(\mathbb{r}_1^{\boldsymbol{\mathcal{D}}_1}, \mathbb{r}_2^{\boldsymbol{\mathcal{D}}_2}). \tag{5.43}$$

- If Q_1 and Q_2 are $a_\emptyset$ and $b_\emptyset$, then

$$\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} a \leftrightarrow b. \tag{5.44}$$

- If $Q_1 = Q_2$ and $\boldsymbol{\mathcal{D}}_1 = \boldsymbol{\mathcal{D}}_2$, then

$$\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} 1. \tag{5.45}$$

- If Q_1 is $Q_{11}\,\mathrm{op}\,Q_{12}$ and Q_2 is $Q_{21}\,\mathrm{op}\,Q_{22}$ where op in both RA-expressions is either of $\cap$, $\cup$, $\bowtie$, then

$$\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} \big(E(Q_{11}, Q_{21})\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \otimes \big(E(Q_{12}, Q_{22})\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2). \tag{5.46}$$

- If Q_1 is $\mathrm{op}(Q_1')$ and Q_2 is $\mathrm{op}(Q_2')$ where op in both RA-expressions is π_R or ρ_f, then

$$\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} \big(E(Q_1', Q_2')\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2). \tag{5.47}$$

- If Q_1 is $Q_{11}\,\mathrm{op}^{Q_{13}}\,Q_{12}$ and Q_2 is $Q_{21}\,\mathrm{op}^{Q_{23}}\,Q_{22}$ where op is $\rightarrowtriangle$ or $\div$, then

$$\begin{aligned}&\big(E(Q_1, Q_2)\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=}\\ &\big(E(Q_{11}, Q_{21})\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \otimes \big(E(Q_{12}, Q_{22})\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2) \otimes \big(E(Q_{13}, Q_{23})\big)(\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2).\end{aligned} \tag{5.48}$$

- If Q_1 is $\sigma_{y\approx d_1}(Q_1')$ and Q_2 is $\sigma_{y\approx d_2}(Q_2')$ where $d_1, d_2 \in D_y$, then

$$\begin{aligned}&\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=}\\ &\big(E(Q_1',Q_2')\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2)\otimes\textstyle\bigwedge_{d\in D_y}\big(d\approx_y^{\boldsymbol{\mathcal{D}}_1} d_1 \leftrightarrow d\approx_y^{\boldsymbol{\mathcal{D}}_2} d_2\big).\end{aligned} \tag{5.49}$$

- If Q_1 is $\sigma_{y\approx y'}(Q_1')$ and Q_2 is $\sigma_{y\approx y'}(Q_2')$ where y, y' are attributes with the same domain, then

$$\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=} \big(E(Q_1',Q_2')\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2)\otimes E\big(\approx_y^{\boldsymbol{\mathcal{D}}_1},\approx_y^{\boldsymbol{\mathcal{D}}_2}\big). \tag{5.50}$$

- If Q_1 is $Q_{11}\bowtie_{y_1\approx y_2} Q_{12}$ and Q_2 is $Q_{21}\bowtie_{y_1\approx y_2} Q_{22}$, then

$$\begin{aligned}&\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=}\\ &\big(E(Q_{11},Q_{21})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2)\otimes\big(E(Q_{12},Q_{22})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2)\otimes E\big(\approx_y^{\boldsymbol{\mathcal{D}}_1},\approx_y^{\boldsymbol{\mathcal{D}}_2}\big).\end{aligned} \tag{5.51}$$

- If Q_1 is $Q_{11}\bowtie_{a/y_1\approx y_2} Q_{12}$ and Q_2 is $Q_{21}\bowtie_{b/y_1\approx y_2} Q_{22}$, then

$$\begin{aligned}&\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \stackrel{\text{def}}{=}\\ &(a\leftrightarrow b)\otimes\big(E(Q_{11},Q_{21})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2)\otimes\big(E(Q_{12},Q_{22})\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2)\otimes E\big(\approx_y^{\boldsymbol{\mathcal{D}}_1},\approx_y^{\boldsymbol{\mathcal{D}}_2}\big).\end{aligned} \tag{5.52}$$

The following theorem shows that similarities as defined above are indeed lower bounds of similarities of query results.

Theorem 50. *Let Q_1 and Q_2 be RA-expressions such that $E(Q_1,Q_2)$ is defined. Then, for any database instances $\boldsymbol{\mathcal{D}}_1$ and $\boldsymbol{\mathcal{D}}_2$, we have*

$$\big(E(Q_1,Q_2)\big)(\boldsymbol{\mathcal{D}}_1,\boldsymbol{\mathcal{D}}_2) \le E\big(Q_1^{\boldsymbol{\mathcal{D}}_1},Q_2^{\boldsymbol{\mathcal{D}}_2}\big). \tag{5.53}$$

5.3 Tuple-based similarity

In this section, we show an alternative definition of similarity of RDTs, which is connected to the notion of similarity-based closure.

While the rank-based similarity (4.1) can be sufficient in many cases there are situations in which the use of (4.1) seems to be inadequate. For example, take the RDT from Section 1.2.3, increase the price of every room by 1 euro and keep all other data and ranks unaltered. Then according to rank-based similarity, the original data table and the new one are very different, their similarity degree will be 0 for any choice of $\mathbf{L}$. Intuitively, since the two data tables differ only by a small change in price, one would expect to have a high degree of similarity. Hence, we wish to consider the values in tuples in addition to the ranks of tuples in RDTs when assessing similarity. Naturally, $\mathcal{D}_1$ and $\mathcal{D}_2$ will likely be considered similar if they pass a test given by the following proposition:

> *For every tuple in $\mathcal{D}_1$, there exists a similar tuple in $\mathcal{D}_2$*
> *and for every tuple in $\mathcal{D}_2$, there exists a similar tuple in $\mathcal{D}_1$.*

That is, one may define

$$S^{\approx}(\mathcal{D}_1, \mathcal{D}_2) = \bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \to \bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}_2(r') \otimes r \approx_R r'\big)\big), \tag{5.54}$$

$$E^{\approx}(\mathcal{D}_1, \mathcal{D}_2) = S^{\approx}(\mathcal{D}_1, \mathcal{D}_2) \wedge S^{\approx}(\mathcal{D}_2, \mathcal{D}_1), \tag{5.55}$$

where the similarity degree $r \approx_R r'$ of tuples r and r' is defined by

$$r \approx_R r' = \bigwedge_{y \in R} r(y) \approx_y r'(y). \tag{5.56}$$

5.3.1 Similarity-based semijoins and closures

The similarity of RDTs based on (5.54) can be expressed using (4.2) and a derived relational operation *similarity closure*, which is a special case of similarity-based semijoin [17]. For nonranked $\mathcal{D}_1$ such that $\mathcal{D}_2 \subseteq \mathcal{D}_1$, *similarity closure of* $\mathcal{D}_2$ (*with respect to* $\mathcal{D}_1$) is defined as:

$$(\mathrm{C}^{\approx}_{\mathcal{D}_1}(\mathcal{D}_2))(r) = \mathcal{D}_1(r) \otimes \bigvee_{r' \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_2(r') \otimes r' \approx_R r\big) \tag{5.57}$$

for each $r \in \mathrm{Tupl}(R_2)$. Furthermore, since $\mathcal{D}_1$ is nonranked, we may write

$$(\mathrm{C}^{\approx}_{\mathcal{D}_1}(\mathcal{D}_2))(r) = \bigvee_{r' \in \mathrm{Tupl}(R_2)} \big(\mathcal{D}_2(r') \otimes r' \approx_R r\big), \tag{5.58}$$

whenever $r \in \mathcal{D}_1$ (and $= 0$ otherwise).

Taking into account (5.57), $\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)$ can be seen as a result of query: "Show all tuples which are in $\mathcal{D}_1$ and, in addition, include all tuples which are from $\mathcal{D}$ and are similar to those in $\mathcal{D}_1$." Similarity-based closures and semijoins may be considered as examples of nontrivial relational operations which do not appear in the classical relational model.

Using similarity-closures, $S^{\approx}$ defined by (5.54) can be restated as

$$\begin{aligned} S^{\approx}(\mathcal{D}_1, \mathcal{D}_2) &= \bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \to \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)(r)\big) \\ &= S\big(\mathcal{D}_1, \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)\big), \end{aligned} \tag{5.59}$$

where $\mathcal{D}$ is nonranked such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$. Clearly, the value of (5.59) does not depend on the choice of a nonranked $\mathcal{D}$ satisfying $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$.

If $\approx_R$ is $\otimes$-transitive, then $\mathrm{C}^{\approx}_{\mathcal{D}}$ forms an **L**-closure operator [4, 3]:

Lemma 51. *Let $\mathcal{D}$ be a nonranked table on R and let $\approx_R$ be $\otimes$-transitive. Then, $\mathrm{C}^{\approx}_{\mathcal{D}}$ is an* **L***-closure operator, i.e., it satisfies*

$$\mathcal{D}_1 \subseteq \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1), \tag{5.60}$$

$$S(\mathcal{D}_1, \mathcal{D}_2) \leq S(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1), \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2)), \tag{5.61}$$

$$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)) = \mathrm{C}^{\approx}_{\mathcal{D}}(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)), \tag{5.62}$$

for all $\mathcal{D}_1$ and $\mathcal{D}_2$ on R such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$.

Based on our observations, we may view $S^{\approx}$ and $E^{\approx}$ as being defined using (4.2), (4.1), and similarity-based closures of RDTs. The following theorem shows further properties of $\mathrm{C}^{\approx}_{\mathcal{D}}$ with respect to other relational operations.

Theorem 52. *For any RDTs $\mathcal{D}_1, \mathcal{D}_2$ and nonranked $\mathcal{D}$ on R, such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$:*

$$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \cup \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2) = \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1 \cup \mathcal{D}_2), \tag{5.63}$$

$$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1 \cap \mathcal{D}_2) \subseteq \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \cap \mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_2), \tag{5.64}$$

If $\mathcal{D}_1$ and $\mathcal{D}_2$ are RDTs on disjoint schemes R_1 and R_2, respectively, $R \subseteq R_1$, then

$$\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1) \bowtie \mathrm{C}^{\approx}_{\mathcal{D}'}(\mathcal{D}_2) \subseteq \mathrm{C}^{\approx}_{\mathcal{D}\bowtie\mathcal{D}'}(\mathcal{D}_1 \bowtie \mathcal{D}_2), \tag{5.65}$$

$$\pi_R(\mathrm{C}^{\approx}_{\mathcal{D}}(\mathcal{D}_1)) \subseteq \mathrm{C}^{\approx}_{\pi_R(\mathcal{D})}(\pi_R(\mathcal{D}_1)), \tag{5.66}$$

for any nonranked $\mathcal{D}$ and $\mathcal{D}'$ on R_1 and R_2, respectively, such that $\mathcal{D}_1 \subseteq \mathcal{D}$ and $\mathcal{D}_2 \subseteq \mathcal{D}'$.

5.3.2 Tuple-based similarity estimates

As in case of the rank-based similarity introduced in Section 4.1, we may investigate inequalities which provide tuple-based similarity estimates of query results based on input data. Unlike the rank-based approach, the tuple-based approach has some limitations. In this section, we provide an if and only if criterion for general relational operations which preserve tuple-based similarity.

In the section, we make the following assumptions. We consider relation schemes $R_1, \ldots, R_n, R$ and a map f which maps any RDTs $\mathcal{D}_1, \ldots, \mathcal{D}_n$ on $R_1, \ldots, R_n$ to an RDT $f(\mathcal{D}_1, \ldots, \mathcal{D}_n)$ on R (called the result of f). The map f represents a general n-ary relational operation for which we investigate the issues related to preservation of tuple-based similarity.

Furthermore, let $\odot$ be a binary operation on L with 1 being its neutral element. The operation f is called *S-compatible with respect to $\odot$* if for some $0 \leq j \leq n$, we have

$$\bigodot_{i=1}^{j} S(\mathcal{D}_i, \mathcal{D}'_i) \odot \bigodot_{i=j+1}^{n} S(\mathcal{D}'_i, \mathcal{D}_i) \leq S\big(f(\mathcal{D}_1, \ldots, \mathcal{D}_n), f(\mathcal{D}'_1, \ldots, \mathcal{D}'_n)\big), \tag{5.67}$$

for all $\mathcal{D}_i, \mathcal{D}'_i$ on R_i $(i = 1, \ldots, n)$. Analogously, f is called *$S^{\approx}$-compatible with respect to $\odot$* if (5.67) holds for S replaced by $S^{\approx}$. Furthermore, f is called *E-compatible* and *$E^{\approx}$-compatible* if (5.67) holds for S replaced by E and $E^{\approx}$, respectively.

Theorem 53. *Let f be S-compatible with respect to $\odot$. Then, the following statements are equivalent:*

(i) For any $\mathcal{D}_1, \ldots, \mathcal{D}_n$ and nonranked $\mathcal{D}'_1, \ldots, \mathcal{D}'_n$ such that $\mathcal{D}_1 \subseteq \mathcal{D}'_1, \ldots, \mathcal{D}_n \subseteq \mathcal{D}'_n$ there is a nonranked $\mathcal{D}$ such that $f(\mathcal{D}_1, \ldots, \mathcal{D}_n) \subseteq \mathcal{D}$ and

$$f\big(\mathrm{C}^{\approx}_{\mathcal{D}'_1}(\mathcal{D}_1), \ldots, \mathrm{C}^{\approx}_{\mathcal{D}'_n}(\mathcal{D}_n)\big) \subseteq \mathrm{C}^{\approx}_{\mathcal{D}}(f(\mathcal{D}_1, \ldots, \mathcal{D}_n));$$

(ii) f is $S^{\approx}$-compatible with respect to $\odot$.

Theorem 53 enables us to simplify proofs for $S^{\approx}_{\mathcal{D}}$-compatibility of relational operations. In order to prove that operation f is $S^{\approx}$-compatible, it is sufficient to show that f is S-compatible together with (i) of Theorem 53.

Corollary 54. *The following inequalities*

$$S^{\approx}(\mathcal{D}_1, \mathcal{D}_2) \leq S^{\approx}(\pi_R(\mathcal{D}_1), \pi_R(\mathcal{D}_2)),$$
$$S^{\approx}(\mathcal{D}_1, \mathcal{D}_1') \wedge S^{\approx}(\mathcal{D}_2, \mathcal{D}_2') \leq S^{\approx}(\mathcal{D}_1 \cup \mathcal{D}_2, \mathcal{D}_1' \cup \mathcal{D}_2'),$$
$$S^{\approx}(\mathcal{D}_1, \mathcal{D}_2) \otimes S^{\approx}(\mathcal{D}_3, \mathcal{D}_4) \leq S^{\approx}(\mathcal{D}_1 \bowtie \mathcal{D}_3, \mathcal{D}_2 \bowtie \mathcal{D}_4),$$

hold for any RDTs provided that the relations schemes of $\mathcal{D}_1$ and $\mathcal{D}_3$ ($\mathcal{D}_2$ and $\mathcal{D}_4$) are disjoint. Moreover, the same inequalities hold if $S^{\approx}$ is replaced by $E^{\approx}$. □

It can be shown by means of simple counterexamples that relational operations from previous sections excluding those listed in Corollary 54 are not $S^{\approx}$-compatible with respect to $\otimes$.

5.3.3 Unifying approach to similarity of RDTs

It was shown in [6] that both (4.2) and (5.59) have a common generalization using truth-stressing hedge. Let * be truth-stressing hedge on $\mathbf{L}$. For RDTs $\mathcal{D}_1, \mathcal{D}_2$ on R and nonranked $\mathcal{D}$ on R such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$ we define a degree $S_*^{\approx}(\mathcal{D}_1, \mathcal{D}_2)$ of inclusion of $\mathcal{D}_1$ in $\mathcal{D}_2$ (with respect to *) and a degree of similarity $E_*^{\approx}(\mathcal{D}_1, \mathcal{D}_2)$ with respect to * as

$$S_*^{\approx}(\mathcal{D}_1, \mathcal{D}_2) = \textstyle\bigwedge_{r \in \mathrm{Tupl}(R)} \big(\mathcal{D}_1(r) \to \mathrm{C}_{\mathcal{D}}^{\approx^*}(\mathcal{D}_2)\big), \tag{5.68}$$
$$E_*^{\approx}(\mathcal{D}_1, \mathcal{D}_2) = S_*^{\approx}(\mathcal{D}_1, \mathcal{D}_2) \wedge S_*^{\approx}(\mathcal{D}_2, \mathcal{D}_1), \tag{5.69}$$

where $\mathrm{C}_{\mathcal{D}}^{\approx^*}(\mathcal{D}_2)$ is a similarity-based closure of $\mathcal{D}_2$ with respect to $\mathcal{D}$ and hedge $*$ and is defined as

$$(\mathrm{C}_{\mathcal{D}}^{\approx^*}(\mathcal{D}_2))(r) = \mathcal{D}(r) \otimes \textstyle\bigvee_{r' \in \mathrm{Tupl}(R)} \big(\mathcal{D}_2(r') \otimes (r' \approx r)^*\big). \tag{5.70}$$

If there is no confusion, we will denote $\mathrm{C}_{\mathcal{D}}^{\approx^*}(\mathcal{D}_2)$ by $\mathrm{C}_{\mathcal{D}}^{*}(\mathcal{D}_2)$.

Now, observe that for $*$ being the identity, (5.68) coincides with (5.54). Furthermore if $\approx_R$ is separating (i.e., $r_1 \approx r_2 = 1$ iff r_1 is identical to r_2) and $*$ is the globalization, (5.68) coincides with (4.2). Thus, both rank-based similarity (4.1) and tuple-based similarity (5.55) are particular instances of (5.69).

By considering two different hedges $*_1, *_2$ on $\mathbf{L}$ we obtain for any RDTs two different subsethood degrees (and two similarity degrees), one using $*_1$ and one using $*_2$. We will denote such degree $S_{*_1}^{\approx}$ and $S_{*_2}^{\approx}$.

Theorem 55. *Let $*_1, *_2$ be two hedges on $\mathbf{L}$. Then for any RDTs $\mathcal{D}_1$, $\mathcal{D}_2$ and any nonranked RDT $\mathcal{D}$ on R such that $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$ we have:*

$$(*_1 \preceq *_2) \leq S_{*_1}^{\approx}(\mathcal{D}_1, \mathcal{D}_2) \to S_{*_2}^{\approx}(\mathcal{D}_1, \mathcal{D}_2), \tag{5.71}$$
$$(*_1 \approx *_2) \leq E_{*_1}^{\approx}(\mathcal{D}_1, \mathcal{D}_2) \leftrightarrow E_{*_2}^{\approx}(\mathcal{D}_1, \mathcal{D}_2). \tag{5.72}$$

In words, (5.71) says that if $\mathcal{D}_1$ is a subset of $\mathcal{D}_2$ using $*_1$ and if $*_1$ is stronger than $*_2$, then $\mathcal{D}_1$ is a subset of $\mathcal{D}_2$ using $*_2$. Analogously, (5.72) says that if hedges $*_1$ and $*_2$ are similar, then the degree of similarity of $\mathcal{D}_1$ and $\mathcal{D}_2$ using $*_1$ and $*_2$ are similar.

Chapter 6

Conclusions

We have studied a particular rank-aware relational model over domains with similarities, which was introduced by Belohlavek and Vychodil.

First, we have presented various extensions of the relational model, which have one in common: the equality relation is replaced by similarity relation. We have focused on generalizations of functional dependencies (GFDs) and established a criterion which makes the comparison of various generalizations easier and more objective. The criterion gives us a degree to which a particular generalization of FD corresponds to the fuzzy function. We have observed that although the definition of fuzzy function is natural and widely accepted, many approaches to GFD failed to satisfy the criterion. One reason is that the validity of GFD usually remains crisp.

Second, we have studied the similarity-based functional dependencies (SBFDs) proposed by Belohlavek and Vychodil and we have presented a graph-based inference methods. We have introduced a notion of a T-based $\mathbf{L}^*$-derivation directed acyclic graph (DAG) which generalizes the ordinary notion of a T-based derivation DAG from [61]. The main results show that degrees of semantic entailment of SBFD from collections of other SBFDs can be characterized by the existence of such directed acyclic graphs.

Third, we have introduced a similarity measure for ranked data tables (RDTs), called rank-based similarity. We have presented several estimates for SBFDs.

Fourth, we have investigated the questions related to similarity preservation. We have shown that if a relational operation (from Section 1.2.3) is applied to pairwise similar input arguments (i.e., pairwise similar RDTs), it produces similar results. In addition, the degree of similarity of the results can be estimated based on the degrees of similarity of the input arguments prior to the evaluation of relational operations, i.e., prior to the execution of a relational query.

Fifth, we have investigated similarity of ranked data tables, based on pairwise similar tuple values. We have focused on the role of similarity-based closures of ranked data tables which are new and nontrivial relational operations. We have shown that tuple-based similarity can be reduced to rank-based similarity of similarity-based closures.

Shrnutí v českém jazyce

Relační databáze, založené na relačním modelu dat (E. F. Codd 1970 [30]), jsou dnes standardem pro ukládání a manipulaci s daty. Za úspěchem relačního modelu stojí, mimo jiné, jeho pevné matematické základy – teorie množin a (dvouhodnotová) predikátová logika. To, co je na jednu stranu výhodou, je na druhou stranu limitující. Relační databáze založené na klasickém relačním modelu neumí pracovat s koncepty, které nejsou bivalentní, ale vícehodnotové, např. s podobností.

Představme si, že hledáme hotel v Olomouci, který nabízí pokoje za 100 €. Klasické relační databáze nám vrátí množinu hotelů, jejichž cena je přesně 100 €. Je ale přirozené, že vedle hotelů stojících 100 € nás zajímají i hotely, jejichž ceny jsou blízko naší představě (např. hotely s cenou 95 € nebo 105 €). Na úvaze se nic nezmění, budeme-li hledat hotely s cenou v nějakém intervalu, např. 95–105 €. Opět nás budou zcela určitě zajímat i hotely, jejichž cena je dostatečně blízko (je podobná) našim požadavkům, např. hotely s cenou 89 € nebo 110 €.

Snahy rozšířit relační model o podobnosti na doménách (doména je množina možných hodnot pro daný atribut) se objevují už od roku 1982 [24]. Podobnost na doméně D_y atributu y lze formalizovat pomocí binární fuzzy relace $\approx_y$: $D_y \times D_y \to L$. Tedy každým dvěma hodnotám $d_1, d_2 \in D_y$ je přiřazen stupeň jejich podobnosti $(d_1 \approx_y d_2) \in L$. Často se volí $L = [0, 1]$. Relačním modelům, které uvažují podobnosti na doménách, budeme říkat podobnostní relační modely.

Disertační práce je věnována relačnímu modelu dat, který představili Bělohlávek a Vychodil [12], a který rozšiřuje původní relační model takto: 1) Na každé doméně je zavedena relace podobnosti. 2) Relace (databázové tabulky) jsou rozšířené o tzv. ranky. Každý řádek (záznam) obsahuje navíc rank, což je stupeň, ve kterém daný řádek vyhovuje dotazu. Tento model je založen na predikátové fuzzy logice.

První část disertační práce je zaměřena na funkční závislosti v podobnostních relačních modelech, které se snaží popsat závislosti typu: Jestliže jsou si dva řádky podobné na atributech A, pak jsou si podobné na atributech B. Přístupů k funkčním závislostem v podobnostních relačních modelech je několik desítek, pozornost je proto věnována porovnání těchto přístupů. Je představeno kritérium, které umožňuje rozdílné definice objektivně srovnat. U funkčních závislostí, které představili Bělohlávek a Vychodil, jsou A, B fuzzy množiny atributů. Příkladem takové závislosti může být: Jestliže mají hotely podobnou cenu alespoň ve stupni 0,8, pak mají podobné hodnocení od zákazníků alespoň ve stupni 0,7. Formálně lze psát $\{^{0,8}/cena\} \Rightarrow \{^{0,7}/\text{hodnocení}\}$. Pravdivost funkčních závislostí se uvažuje ve stupních. V disertační práci je pro tyto funkční závislosti vyvinut alternativní dokazovací systém, který je založen na orientovaných grafech. Je dokázána úplnost

v následujícím smyslu: Funkční závislost $A \Rightarrow B$ sémanticky plyne z množiny funkčních závislostí tehdy a jen tehdy, existuje-li orientovaný graf pro $A \Rightarrow B$. Konstrukci orientovaných grafů lze využít i pro určení uzávěru (fuzzy) množiny atributů vzhledem k teorii.

Druhá část práce je věnována citlivosti funkčních závislosti a relačních operací (v modelu Bělohlávka a Vychodila) na vstupních datech. Nejprve je diskutováno, jak lze měřit podobnost databázových tabulek (relací s ranky) a jsou představeny dvě míry: podobnost založená na rancích (rank-based similarity) a podobnost založená na datech (tuple-based similarity). U podobnosti založené na rancích řekneme, že dvě relace s ranky jsou si podobné, pokud stejné řádky patří do obou relací v podobném stupni. Pro tuto podobnost je dokázáno, že v podobných relacích budou funkční závislosti platit v podobném stupni. Tedy, že definice funkčních závislostí je robustní: malá změna na vstupních datech způsobí pouze malou změnu v platnosti funkčních závislostí. Rovněž jsou prezentovány odhady pro pravdivost funkční závislosti $A_1 \Rightarrow B_1$, $A_2 \Rightarrow B_2$ v závislosti na podobnosti fuzzy množin atributů A_1, A_2 a B_1, B_2. Pro podobnost založenou na rancích je dále studována citlivost výsledků relačních operací na vstupních datech. Je ukázáno, že pro libovolný dotaz lze podobnost výsledků dotazu odhadnout na základě podobnosti vstupních dat.

U podobnosti založené na datech řekneme, že dvě relace s ranky jsou si podobné, jestliže ke každému řádku v jedné relaci existuje řádek v druhé relaci, který je mu podobný a opačně. Ukazuje se, že tuto podobnost lze vyjádřit pomocí podobnosti založené na rancích a nové relační operace: podobnostního uzávěru. V disertační práci jsou studovány vlastnosti podobnostního uzávěru a jeho vztah k relačním operacím. Rovněž je představena podobnost pro relace s ranky, která zobecňuje obě předchozí.

Selected publications of the author

- L. Ježková, P. Cordero, M. Enciso: *Codd's Relational Model of Data Over Domains With Similarities: A Comparative Survey*, Fuzzy Sets and Systems, submitted
- L. Urbanová and V. Vychodil: *Derivation digraphs for dependencies in ordinal and similarity-based data*, Information Sciences 268 (2014), pp. 381-–396
- R. Bělohlávek, L. Urbanová and V. Vychodil: *Sensitivity Analysis for Declarative Relational Query Languages with Ordinal Ranks*, In: Tompits H., Abreu S., Oetsch J., Pührer J., Seipel D., Umeda M., Wolf A. (Eds.): Applications of Declarative Programming and Knowledge Management: 19th International Conference, INAP 2011, Lecture Notes in Artificial Intelligence 7773, 2013, pp. 58–76
- L. Urbanová, V. Vychodil and L. Wiese: *Applications of Ordinal Ranks to Flexible Query Answering*, In: Hüllermeier E., Link S., Fober T., Seeger B. (Eds.): Scalable Uncertainty Management: 6th International Conference, Lecture Notes in Computer Science 7520, 2012, pp. 16—29
- R. Bělohlávek, L. Urbanová and V. Vychodil: *Similarity of query results in similarity-based databases*, In: Yao J. T., Ramanna S., Wang G., Suraj Z. (Eds.): Rough Sets and Knowledge Technology, Lecture Notes in Computer Science 6954, 2011, pp. 258—267

Bibliography

[1] F. E. Petry B. P. Buckles and H. S. Sachar. Design of similarity-based relational databases. In Constantin V. Negoita Henri Prade, editor, *Fuzzy logic in knowledge engineering*, pages 3–17. TUV Rheinland, 1986.

[2] J.F. Baldwin and S.Q Zhou. A fuzzy relational inference language. *Fuzzy Sets and Systems*, 14(2):155 – 174, 1984.

[3] R. Belohlavek. Fuzzy closure operators. *Journal of Mathematical Analysis and Applications*, 262(2):473 – 489, 2001.

[4] R. Belohlavek. *Fuzzy Relational Systems: Foundations and Principles*. Kluwer Academic Publishers, 2002.

[5] R. Belohlavek, L. Urbanova, and V. Vychodil. Similarity of query results in similarity-based databases. In Jing-Tao Yao, Sheela Ramanna, Guoyin Wang, and Zbigniew Suraj, editors, *Rough Sets and Knowledge Technology*, volume 6954 of *Lecture Notes in Computer Science*, pages 258–267. Springer Berlin Heidelberg, 2011.

[6] R. Belohlavek, L. Urbanova, and V. Vychodil. Sensitivity analysis for declarative relational query languages with ordinal ranks. In H. Tompits, S. Abreu, J. Oetsch, J. Pührer, D. Seipel, M. Umeda, and A. Wolf, editors, *Applications of Declarative Programming and Knowledge Management*, volume 7773 of *Lecture Notes in Computer Science*, pages 58–76. Springer Berlin Heidelberg, 2013.

[7] R. Belohlavek and V. Vychodil. Attribute implications in a fuzzy setting. In R. Missaoui and J. Schmidt, editors, *Formal Concept Analysis*, volume 3874 of *Lecture Notes in Computer Science*, pages 45–60. Springer Berlin / Heidelberg, 2006.

[8] R. Belohlavek and V. Vychodil. Axiomatization of fuzzy attribute logic over complete residuated lattices. In *Proceedings of the 2006 Joint Conference on Information Sciences, JCIS 2006, Kaohsiung, Taiwan, ROC, October 8-11*, 2006.

[9] R. Belohlavek and V. Vychodil. Data tables with similarity relations: Functional dependencies, complete rules and non-redundant bases. In *Proceedings of the 11th International Conference on Database Systems for Advanced Applications*, DASFAA'06, pages 644–658, Berlin, Heidelberg, 2006. Springer-Verlag.

[10] R. Belohlavek and V. Vychodil. Similarity issues in attribute implications from data with fuzzy attributes. In *Information Reuse and Integration, 2006 IEEE International Conference on*, pages 132–135, Sept 2006.

[11] R. Belohlavek and V. Vychodil. On proofs and rule of multiplication in fuzzy attribute logic. In P. Melin, O. Castillo, L. Aguilar, J. Kacprzyk, and W. Pedrycz, editors, *Foundations of Fuzzy Logic and Soft Computing*, volume 4529 of *Lecture Notes in Computer Science*, pages 471–480. Springer Berlin / Heidelberg, 2007.

[12] R. Belohlavek and V. Vychodil. Data dependencies in codd's relational model with similarities. In José Galindo, editor, *Handbook of Research on Fuzzy Information Processing in Databases*, pages 634–657. IGI Global, 2008.

[13] R. Belohlavek and V. Vychodil. Logical foundations for similarity-based databases. In Lei Chen, Chengfei Liu, Qing Liu, and Ke Deng, editors, *Database Systems for Advanced Applications*, volume 5667 of *Lecture Notes in Computer Science*, pages 137–151. Springer Berlin Heidelberg, 2009.

[14] R. Belohlavek and V. Vychodil. Query systems in similarity-based databases: Logical foundations, expressive power, and completeness. In *Proceedings of the 2010 ACM Symposium on Applied Computing*, SAC '10, pages 1648–1655, New York, NY, USA, 2010. ACM.

[15] R. Belohlavek and V. Vychodil. Codd's relational model from the point of view of fuzzy logic. *Journal of Logic and Computation*, 21:851–862, 2011.

[16] R. Belohlavek and V. Vychodil. Relational algebra for multi-ranked similarity-based databases. In *IEEE Symposium on Foundations of Computational Intelligence, FOCI 2013, Singapore, Singapore, April 16-19, 2013*, pages 1–8. IEEE, 2013.

[17] R. Belohlavek and V. Vychodil. Relational similarity-based databases, part 1: Foundations and query systems. *Submitted*, 2014.

[18] R. Belohlavek and V. Vychodil. Relational similarity-based databases, part 2: Dependencies in data. *Submitted*, 2014.

[19] F. Berzal, I. Blanco, D. Sánchez, J. M. Serrano, and M. A. Vila. A definition for fuzzy approximate dependencies. *Fuzzy Sets Syst.*, 149(1):105–129, January 2005.

[20] B. Bhuniya and P. Niyogi. Lossless join property in fuzzy relational databases. *Data & Knowledge Engineering*, 11(2):109–124, 1993.

[21] P. Bosc, D. Dubois, and H. Prade. Fuzzy functional dependencies - an overview and a critical discussion. In *Proceedings of the Third IEEE International Conference on Fuzzy Systems*, pages 325–330, 1994.

[22] P. Bosc, D. Dubois, and H. Prade. Fuzzy functional dependencies and redundancy elimination. *J. Am. Soc. Inf. Sci.*, 49:217–235, March 1998.

[23] P. Bosc, O. Pivert, and L. Ughetto. Database mining for the discovery of extended functional dependencies. In *Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American*, pages 580 –584, jul 1999.

[24] B. P. Buckles and F. E. Petry. A fuzzy representation of data for relational databases. *Fuzzy Sets and Systems*, 7(3):213 – 226, 1982.

[25] B. P. Buckles and F. E. Petry. Extending the fuzzy database with fuzzy numbers. *Information Sciences*, 34(2):145 – 155, 1984.

[26] G. Chen. Fuzzy functional dependencies and a series of design issues of fuzzy relational databases. In *Fuzzines in Database Management Systems*, pages 166–185. Physica Verlag, Heidelberg, 1995.

[27] G. Chen, E. E. Kerre, and J. Vandenbulcke. A computational algorithm for the ffd transitive closure and a complete axiomatization of fuzzy functional dependencies. *International Journal of Intelligent Systems*, 9:421–439, 1994.

[28] G.Q. Chen. A step towards the theory of fuzzy relational database design. In *Proc. of IFSA'91 World Congress*, pages 44–47, 1991.

[29] R. Cignoli, F. Esteva, L. Godo, and A. Torrens. Basic fuzzy logic is the logic of continuous t-norms and their residua. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, 4:106–112, 2000.

[30] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, June 1970.

[31] E. F. Codd. Extending the database relational model to capture more meaning. *ACM Trans. Database Syst.*, 4(4):397–434, December 1979.

[32] E. F. Codd. More commentary on missing information in relational databases (applicable and inapplicable information). *SIGMOD Rec.*, 16(1):42–50, March 1987.

[33] P. Cordero, M. Enciso, A. Mora, and I. Perez de Guzman. A complete axiomatic system for fuzzy functional dependencies over domains with similarity relations. *Lecture Notes Computer Science - IWANN 09*, 5517:261–269, 2009.

[34] P. Cordero, M. Enciso, A. Mora, I. Perez de Guzman, and J.M. Rodriguez-Jimenez. Specification and inference of fuzzy attributes. In *Foundations of Computational Intelligence (FOCI), 2011 IEEE Symposium on*, pages 107–114, 2011.

[35] J. C. Cubero, J. M. Medina, O. Pons, and M. A. Vila. Non-transitive fuzzy dependencies (i). *Fuzzy Sets Syst.*, 106:401–431, September 1999.

[36] J. C. Cubero and M. A. Vila. A new definition of fuzzy functional dependency in fuzzy relational databases. *International Journal of Intelligent Systems*, 9(5):441–448, 1994.

[37] C. J. Date. *Relational Database:Selected Writings*. Addison Wesley Publishing Company, 1986.

[38] C. J. Date. *Date on Database: Writings 2000–2006*. Apress, 2006.

[39] C. J. Date and H. Darwen. *Databases, Types and the Relational Model (3rd Edition)*. Addison-Wesley, 2006.

[40] M. Demirci. Fuzzy functions and their applications. *Journal of Mathematical Analysis and Applications*, 252(1):495 – 517, 2000.

[41] D. Dubois and H. Prade. Certainty and uncertainty of (vague) knowledge and generalized dependencies in fuzzy databases. In *Fuzzy Engineering Toward Human Friendly Systems*, pages 239–249. IOS Press, 1992.

[42] F. Esteva, L. Godo, and C. Noguera. A logical approach to fuzzy truth hedges. *Information Sciences*, 232:366–385, 2013.

[43] R. Fagin. Fuzzy queries in multimedia database systems. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '98, pages 1–10, New York, NY, USA, 1998. ACM.

[44] W. Fan, H. Gao, X. Jia, J. Li, and S. Ma. Dynamic constraints for record matching. *The VLDB Journal*, 20(4):495–520, August 2011.

[45] J. Vandenbulcke G. Q. Chen and E. E. Kerre. Fuzzy functional dependency and its axiomatic system in a fuzzy relational data model. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty (IPMU)*, pages 313–316, 1992.

[46] S. Gottwald. Fuzzy uniqueness of fuzzy mappings. *Fuzzy Sets and Systems*, 3(1):49 – 74, 1980.

[47] S. Gottwald. Mathematical fuzzy logics. *Bulletin of Symbolic Logic*, 14:210–239, 6 2008.

[48] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *Proceedings of the Twenty-sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '07, pages 31–40, New York, NY, USA, 2007. ACM.

[49] M. Hajdinjak and G. Bierman. Extending relational algebra with similarities. *Mathematical. Structures in Comp. Sci.*, 22(4):686–718, August 2012.

[50] P. Hajek. *Metamathematics of Fuzzy Logic*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1998.

[51] P. Hajek. On very true. *Fuzzy Sets and Systems*, 124(3):329–333, 2001.

[52] L. Jezkova, P. Cordero, and M. Enciso. Codd's relational model of data over domains with similarities: A comparative survey. *Submitted*, 2014.

[53] A. Kiss. λ decomposition of fuzzy relational databases. *Annales Univ. Sci. Budapest*, 12:133–142, 1991.

[54] F. Klawonn. Fuzzy points, fuzzy relations and fuzzy functions. In Vilém Novák and Irina Perfilieva, editors, *Discovering the World with Fuzzy Logic*, pages 431–453. Physica-Verlag GmbH, Heidelberg, Germany, Germany, 2000.

[55] W. H. Lee and C. T. Pang. An extension of semantic proximity for fuzzy functional dependencies. In *The 28th North American Fuzzy Information Processing Society Annual Conference (NAFIPS2009)*, 2009.

[56] J.Y. C. Liu and C. H. Huang. Handling missing data in extended possibility-based fuzzy relational databases. In *Innovations in Bio-Inspired Computing and Applications (IBICA), 2012 Third International Conference on*, pages 57 –62, sept. 2012.

[57] W. Y. Liu. Extending the relational model to deal with fuzzy values. *Fuzzy Sets Syst.*, 60:207–212, December 1993.

[58] W. Y. Liu. Constraints on fuzzy values and fuzzy functional dependencies. *Information Sciences*, 78(3-4):303–309, 1994.

[59] W. Y. Liu. Fuzzy data dependencies and implication of fuzzy data dependencies. *Fuzzy Sets Syst.*, 92:341–348, December 1997.

[60] Z. M. Ma, W. J. Zhang, W. Y. Ma, and F. Mili. Data dependencies in extended possibility-based fuzzy relational databases. *International Journal of Intelligent Systems*, 17(3):321–332, 2002.

[61] D. Maier. Minimum covers in relational database model. *Journal of the ACM*, 27(4):664–674, October 1980.

[62] D. Maier. *Theory of Relational Databases*. Computer Science Pr, Rockville, MD, USA, 1983.

[63] J. M. Medina, M. A. Vila, J. C. Cubero, and O. Pons. Towards the implementation of a generalized fuzzy relational database model. *Fuzzy Sets Syst.*, 75:273–289, November 1995.

[64] N. Mouaddib and N. Bonanno. New semantics for the membership degree in fuzzy databases. In *Uncertainty Modeling and Analysis, 1995, and Annual Conference of the North American Fuzzy Information Processing Society. Proceedings of ISUMA - NAFIPS '95., Third International Symposium on*, pages 655 –660, sep 1995.

[65] K. Myszkorowski. Analysis of fuzzy n-ary relations with the use of interval-valued fuzzy functional dependencies. *International Journal of General Systems*, 42, 2013.

[66] M. Nakata. Dependencies in fuzzy databases: functional dependency. In *Proceedings of 1995 IEEE International Conference on Fuzzy Systems*, volume 2, pages 757–764, Yokohama, Japan, 1995.

[67] L. Lietard P. Bosc and O. Pivert. Functional dependencies revisited under graduality and imprecision. In *Fuzzy Information Processing Society, 1997. NAFIPS '97., 1997 Annual Meeting of the North American*, pages 57–62, 1997.

[68] J. Pavelka. On fuzzy logic I: Many-valued rules of inference. *Mathematical Logic Quarterly*, 25(3–6):45–52, 1979.

[69] J. Pavelka. On fuzzy logic II: Enriched residuated lattices and semantics of propositional calculi. *Mathematical Logic Quarterly*, 25(7–12):119–134, 1979.

[70] J. Pavelka. On fuzzy logic III: Semantical completeness of some many-valued propositional calculi. *Mathematical Logic Quarterly*, 25(25–29):447–464, 1979.

[71] H. Prade. Lipski's approach to incomplete information data bases restated and generalized in the setting of zadeh's possibility theory. *Information Systems*, 9(1):27–42, 1984.

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

[73] K. V. S. V. N. Raju and A. K. Majumdar. The study of joins in fuzzy relational databases. *Fuzzy Sets Syst.*, 21(1):19–34, January 1987.

[74] 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 Syst.*, pages 129–166, 1988.

[75] D. Rasmussen and R. R. Yager. Finding fuzzy and gradual functional dependencies with summarysql. *Fuzzy Sets and Systems*, 106(2):131 – 142, 1999.

[76] A. N. Saharia and T. M. Barron. Approximate dependencies in database systems. *Decision Support Systems*, 13(3-4):335–347, March 1995.

[77] P. C. Saxena and B. K. Tyagi. Fuzzy functional dependencies and independencies in extended fuzzy relational database models. *Fuzzy Sets and Systems*, 69(1):65–89, 1995.

[78] A.K. Sharma, A. Goswami, and D.K. Gupta. Fuzzy inclusion dependencies in fuzzy relational databases. In *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, volume 1, pages 507 – 510 Vol.1, april 2004.

[79] S. Shenoi, A. Melton, and L. T. Fan. An equivalence classes model of fuzzy relational databases. *Fuzzy Sets and Systems*, 38(2):153–170, 1990.

[80] M. I. Sozat and A. Yazici. A complete axiomatization for fuzzy functional and multivalued dependencies in fuzzy database relations. *Fuzzy Sets and Systems*, 117:161–181, 2001.

[81] Y. Takahashi. Fuzzy database query languages and their relational completeness theorem. *Knowledge and Data Engineering, IEEE Transactions on*, 5(1):122–125, Feb 1993.

[82] G. Takeuti and S. Titani. Globalization of intuitionistic set theory. *Annals of Pure and Applied Logic*, 33:195–211, 1990.

[83] B. K. Tyagi, A. Sharfuddin, R. N. Dutta, and D. K. Tayal. A complete axiomatization of fuzzy functional dependencies using fuzzy function. *Fuzzy Sets and Systems*, 151:363–379, 2005.

[84] M. Umano. Freedom-o: A fuzzy database system. In Gupta Sanchez, editor, *Fuzzy Information and Decision Processes*, pages 339–347. North-Holand Pub. Comp., 1982.

[85] M. Umano. Retrieval from fuzzy databases by fuzzy relational algebra. In G. Sanchez, editor, *Fuzzy Information Knowledge Representation and Decision Analysis*, pages 1–6. Pergamon Press, Oxford, 1983.

[86] L. Urbanova and V. Vychodil. Derivation digraphs for dependencies in ordinal and similarity-based data. *Information Sciences*, 268(0):381 – 396, 2014. New Sensing and Processing Technologies for Hand-based Biometrics Authentication.

[87] L. Urbanova, V. Vychodil, and L. Wiese. Applications of ordinal ranks to flexible query answering. In *Scalable Uncertainty Management - 6th International Conference, SUM 2012, Marburg, Germany, September 17-19, 2012. Proceedings*, pages 16–29, 2012.

[88] M. Vucetic and M. Vujosevic. A literature overview of functional dependencies in fuzzy relational database models. *Technics Technologies Education Management-TTEM*, 7(4):1593–1604, 2012.

[89] S. L. Wang, J. S. Tsai, and B. C. Chien. Mining approximate dependencies using partitions on similarity-relation-based fuzzy databases. In *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 5, pages 871 –875 vol.5, 1999.

[90] S. Ben Yahia, H. Ounalli, and A. Jaoua. An extension of classical functional dependency: dynamic fuzzy functional dependency. *Information Sciences*, 119(3-4):219 – 234, 1999.

[91] A. Yazici, E. Gocmen, B.P. Buckles, R. George, and F.E. Petry. An integrity constraint for a fuzzy relational database. In *Proc. of Second IEEE Int. Conf. on Fuzzy Systems 1*, pages 496–499, 1993.

[92] A. Yazici and M.I. Sozat. The integrity constraints for similarity-based fuzzy relational databases. *International Journal of Intelligent Systems*, 13:641–659, 1998.

[93] L. A. Zadeh. Fuzzy sets. *Information and control*, 8:338 – 353, 1965.

[94] L. A. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1(1):3 – 28, 1978.

[95] F. Zhao and Z.M. Ma. Functional dependencies in vague relational databases. In *2006 IEEE International Conference on Systems, Man, and Cybernetics*, pages 4006–4010, 2006.

[96] A. Zvieli. A fuzzy relational calculus. In *Expert Database Conf.'86*, pages 311–326, 1986.