Similarity Measures for Relational Databases

Melita Hajdinjak
University of Ljubljana, Faculty of Electrical Engineering, Tržaška 25, 1000 Ljubljana, Slovenia
melita.hajdinjak@fe.uni-lj.si and http://matematika.fe.uni-lj.si/

Andrej Bauer
University of Ljubljana, Faculty of Mathematics and Physics, Jadranska 21, 1000 Ljubljana, Slovenia
andrej.bauer@fmf.uni-lj.si and http://andrej.com/

Keywords: relational algebra, related answers, similarity search

Received: November 5, 2008

We enrich sets with an integrated notion of similarity, measured in a (complete) lattice, special cases of which are reflexive sets *and* bounded metric spaces. *Relations and basic relational operations of traditional relational algebra are interpreted in such richer structured environments. An canonical similarity measure between relations is introduced. In the special case of reflexive sets it is just the well known* Egli-Milner ordering *while in the case of bounded metric spaces it is the* Hausdorff metric. *Some examples of how to perform approximate searches (e.g.,* similarity search *and* relaxed answers*) are given.*

Povzetek: Z željo po iskanju bližnjih informacij in relaksiranih odgovorov množice obogatimo z merami podobnosti. Interpretiramo relacije in operacije relacijske algebre.

1 Introduction

The *relational algebra* (4; 15), a relational data model with five basic operations on relations, i.e., *Cartesian product* $\times$, *projection* π, *selection* σ, *union* $\cup$, and *set difference* $-$, and several additional operations such as *θ-join* or *intersection*, has three main advantages over non-relational data models (13):

- From the point of view of *usability*, the model has a simple interpretation in terms of real-world concepts, i.e., the essential data structure of the model is a relation, which can be visualized in a tabular format.
- From the point of view of *applicability*, the model is flexible and general, and can be easily adapted to many applications.
- From the point of view of *formalism*, the model is elegant enough to support extensive research and analysis.

Hence, the relational data models have gained acceptance from a broad range of users, they have gained popularity and credibility in a variety of application areas, and they facilitate better theoretical research in many fundamental issues arising from database query languages and dependency theory.

However, there are several applications that have evolved beyond the capabilities of traditional relational data models, such as applications that require databases to *cooperate* with the user by suggesting answers which may be helpful but were not explicitly asked for. The *cooperative-behaviour* or *cooperative-answering* techniques (5) may be differentiated into the following categories:

i.) consideration of specific information about a user's state of mind,

ii.) evaluation of presuppositions in a query,

iii.) detection and correction of misconceptions in a query,

iv.) formulation of intensional answers,

v.) generalization of queries and of responses.

The cooperative behaviour plays an important part, for instance, in information-providing dialogue systems (7), where the most vital cooperative-answering technique leading to user satisfaction is *generalization of queries and of responses* as shown by Hajdinjak and Mihelič (8). Generalization of queries and of responses, the aim of which is to capture possibly relevant information, is often achieved by *query relaxation* (6).

Another kind of applications not suitable for the traditional relational data models are applications which require the database to be enhanced with a notion of *similarity* that allows one to perform *approximate* searches (9). The goal in these applications is often one of the following:

i.) Find objects whose feature values fall within a given range or where the distance from some query object falls into a certain range (range queries).

ii.) Find objects whose features have values similar to those of a given query object or set of query objects (nearest neighbour queries and approximate nearest neighbour queries).

iii.) Find pairs of objects from the same set or different sets which are sufficiently similar to each other (closest pairs queries).

Examples of such approximate-matching or similarity-search applications are databases storing images, fingerprints, audio clips or time sequences, text databases with typographical or spelling errors, text databases where we look for documents that are similar to a given document, and computational-biology applications where we want to find a DNA or a protein sequence in a database allowing some errors due to typical variations.

Persuaded that many applications will never reach the limitations of the widespread relational data model this article focuses on traditional relational algebra equipped with extra features that allow query relaxation and similarity searches. Although a large body of work has addressed how to extend the relational data model to incorporate cooperativity, neighbouring information, and/or orderings (2; 3; 10; 11; 13), neither of them have succeeded to fit into the representational and operational uniformity of traditional relational algebra or even to reach a certain degree of generality.

Therefore, we are going to talk about domains, similarity, approximate answers, and nearness of data in a highly systematic and comprehensive way, which will lead us towards an usable, applicable, and a formaly strong generalization of the relational data model.

2 Sets with similarity

Most applications and proposed solutions of non-exact matches and similarity search, which are not covered by traditional relational algebra, have some common characteristics – there is a universe of objects and a non-negative distance or distance-like function defined among them. The distance function measures how close are the non-exact matches to the exact specifications that were given by the user willing to accept approximate answers.

Instead of restricting only to distance metrics, we consider more general *similarity measures* that satisfy the only condition of being *reflexive*, i.e., every object is most similar to itself. Hence, rather than focusing on (ordinary sets) or metric spaces, we will consider more general *sets with similarity*, where a measure of similarity assigns to a pair of objects a similarity value, which tells us how similar they are. Note, we speak of *similarity* instead of distance – if a point x moves toward a point y, the distance between x and y gets smaller, but their similarity gets larger.

For the domain of possible similarity values we choose *complete lattices*, i.e., partially ordered sets in which all subsets have both a least upper bound (*join*) and a greatest lower bound (*meet*).

Definition 1. A *set with similarity* is an ordered triple

$$\underline{A} = (A, L_A, \rho_A),$$

where A is the *underlying set*, L_A is a complete lattice with the least element 0_A and the greatest element 1_A, and

$$\rho_A : A \times A \to L_A$$

is a *measure of similarity* in A satisfying the *reflexivity* condition

$$\rho_A(x, x) = 1_A$$

for all $x \in A$.

In the trivial case, if we take the complete lattice L_2 of boolean values $\{0, 1\}$ equipped with minimum and maximum as the operations meet and join, respectively, ordered with relation $\leq$, and define the similarity by

$$\rho(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{if } x \neq y, \end{cases}$$

the resulting set with similarity gains no additional structure. That is, it is equivalent to the underlying set.

There are many non-trivial similarities and, consequently, non-trivial sets with similarity, such as *reflexive sets* and *bounded metric spaces*.

Definition 2. A *reflexive set* is an ordered pair $(A, \lhd_A)$, where A is the underlying set, and

$$\lhd_A : A \times A \to L_2$$

is a reflexive relation in A. Habitually, instead of $\lhd_A(x, y) = 1$ we write $x \lhd_A y$.

Since $x \lhd_A x$ for all $x \in A$, the reflexive relation $\lhd_A$ can be understood as a special case of a measure of similarity, thus the reflexive set $(A, \lhd_A)$ can be transformed to the set with similarity $(A, L_2, \lhd_A)$ and embedded into sets with similarity.

Definition 3. A *bounded metric space* is an ordered pair (A, d_A), where A is the underlying set, and

$$d_A : A \times A \to [0, \infty]$$

is a distance function, which satisfies the conditions of non-negativity, symmetry, and triangle inequality:

a.) $d(x, y) \geq 0$ (non-negativity)
and $d(x, y) = 0 \iff x = y$,

b.) $d(x, y) = d(y, x)$, (symmetry)

c.) $d(x, y) \leq d(x, y) + d(y, z)$. (triangle inequality)

In an arbitrary metric space the distance is measured by values strictly smaller than ∞. By allowing ∞ as a distance we have in effect restricted to bounded metric spaces. While the modest generalization of allowing ∞ as a similarity value does not pose a serious restriction (databases are usually built from finite, and therefore bounded sets of data), it makes the set $[0, \infty]$, when ordered by the usual $\geq$ relation, a complete lattice $L_{[0,\infty]}$ as required in sets with similarity. Meet and join are computed as supremum and infimum, respectively. Note that we turned $[0, \infty]$ *upside down* so that the least element is ∞ and the greatest is 0.

Hence the metric d_A is again a special case of a measure of similarity because $d_A(x, x) = 0$, which is the greatest element of the complete lattice $L_{[0,\infty]}$. Thus the bounded metric space (A, d_A) can be transformed to the set with similarity $(A, L_{[0,\infty]}, d_A)$ and embedded into sets with similarities.

3 Tables, relations, and basic relational operations

A relational database is composed of several relations in the form of two-dimensional tables of rows and columns containing related tuples. The rows (tuples) are called *records* and the columns (fields in the record) are called *attributes*. Each attribute has a data type that defines the set of possible values. Thus a relation is a subset of a Cartesian product of sets (value domains).

3.1 Cartesian products and subsets

In order to use sets with similarity instead of (ordinary) sets we need a suitable notion of relation between sets with similarity. Hence we first need to know how to interpret Cartesian products and subsets of sets with similarity in a natural and effective way.

Definition 4. The *Cartesian product* of sets with similarity $\underline{A} = (A, L_A, \rho_A)$ and $\underline{B} = (B, L_B, \rho_B)$ is the set with similarity

$$\underline{A} \times \underline{B} = (A \times B, L_A \times L_B, \rho_{A\times B}),$$

where $A \times B$ is the Cartesian product of sets, $L_A \times L_B$ is the product of complete lattices, and the measure of similarity $\rho_{A\times B}$ is given by

$$\rho_{A\times B}((x_1, y_1), (x_2, y_2)) = (\rho_A(x_1, x_2), \rho_B(y_1, y_2)).$$

The corresponding *canonical projections* are $(\pi_1, p_1) : \underline{A} \times \underline{B} \to \underline{A}$ and $(\pi_2, p_2) : \underline{A} \times \underline{B} \to \underline{B}$, where π_1 and π_2 are projections of sets, but p_1 and p_2 are projections of complete lattices.

This interpretation of Cartesian products of sets with similarity is sound since a product of complete lattices is a complete lattice (14) and $\rho_{A\times B}$ satisfies the condition of being a measure of similarity:

$$\begin{aligned} \rho_{A\times B}((x, y), (x, y)) &= (\rho_A(x, x), \rho_B(y, y)) \\ &= (1_A, 1_B) \\ &= 1_{A\times B}, \end{aligned}$$

where 1_A is the greatest element of L_A, 1_B is the greatest element of L_B, and $1_{A\times B}$ is the greatest element of the complete lattice $L_A \times L_B$.

Further, we have decided to consider only those subobjects or substructures $\underline{I}$ of the set with similarity $\underline{A} = (A, L_A, \rho_A)$ whose similarity measure is induced by the structure of $\underline{A}$. That is, the underlying set is a subset of A but the measure of similarity and the corresponding lattice are inherited from $(\underline{A})$. Even though the domain of the measure of similarity has changed from $A \times A$ to $I \times I$, we will keep the notation ρ_A and write $\underline{I} = (I, L_A, \rho_A)$.

Definition 5. (i.e., the Egli-Milner ordering and the Hausdorff metric) A *subset* of the set with similarity $\underline{A} = (A, L_A, \rho_A)$ is a set with similarity $\underline{I} = (I, L_A, \rho_A)$, where $I \subseteq A$. Subsets of sets with similarity will also be called *induced subobjects*.

3.2 Relations and basic relational operations

The family of subsets of $\underline{A}$, denoted by $IndSub(\underline{A})$, is essentially just the power set $\mathcal{P}(A)$.

Theorem 1. The induced subobjects of a set with similarity $\underline{A} = (A, L_A, \rho_A)$ form a complete boolean algebra equivalent to $\mathcal{P}(A)$, in which all the basic relational operations can be properly interpreted.

The formal proof is given in (7). However, the Boolean lattice $IndSub(\underline{A})$ is ordered with the usual subset relation, where the least element is the empty subobject $\underline{\emptyset} = (\emptyset, L_A, \rho_A)$ and the greatest element is $\underline{A}$. Hence selection, union, and difference are calculated as usual $(A_1, A_2 \subseteq A)$:

$$\begin{aligned} &\sigma_F(A_1, L_A, \rho_A) = (\sigma_F(A_1), L_A, \rho_A), \\ &(A_1, L_A, \rho_A) \cup (A_2, L_A, \rho_A) = (A_1 \cup A_2, L_A, \rho_A), \\ &(A_1, L_A, \rho_A) - (A_2, L_A, \rho_A) = (A_1 - A_2, L_A, \rho_A). \end{aligned}$$

Moreover, Cartesian products, projections, selections, unions, and differences of induced subobjects satisfy all the abstract properties that are axiomatized by relational calculus (15; 16).

A relation between two objects of the category of similarities, namely $\underline{A} = (A, L_A, \rho_A)$ and $\underline{B} = (B, L_B, \rho_B)$, is now determined by a subset $R \subseteq A \times B$, which induces a subobject $(R, L_A \times L_B, \rho_{A\times B})$ of the Cartesian product $\underline{A} \times \underline{B}$. Hence tables and answers to queries are modeled as induced subobjects.

4 Similarity of relations

Sets with similarity enjoy additional constructions, which do not exist at the level of underlying sets. For instance, a suitable notion of similarity $\lhd$ between induced subobjects can be defined.

In the case of the reflexive set $(A, \lhd_A)$, which is equipped with a reflexive relation $\lhd_A$ establishing connections between certain elements of A, we propose to take the naturally integrated *Egli-Milner ordering*. Its importance in data models was also recognized by Buneman, Jung, and Ohori (1).

Definition 6. Let $\underline{A_1} = (A_1, \lhd_A)$ and $\underline{A_2} = (A_2, \lhd_A)$ be two induced subobjects of the reflexive set $\underline{A} = (A, \lhd_A)$. The *Egli-Milner ordering* is given as follows:

$$\begin{aligned}\underline{A_1} \lhd \underline{A_2} \quad \iff \;& (\forall x \in A_1\ \exists y \in A_2 : x \lhd_A y) \text{ and} \\ & (\forall y \in A_2\ \exists x \in A_1 : x \lhd_A y).\end{aligned}$$

On the other hand, in the case of the bounded metric space (A, d_A), which is equipped with a distance function d_A, we propose to take the well-known *Hausdorff metric*. It has several applications, for instance, in fractal geometry, in numerical mathematics, and in pattern recognition.

Definition 7. Let $\underline{A_1} = (A_1, d_A)$ and $\underline{A_2} = (A_2, d_A)$ be two induced subobjects of the bounded metric space $\underline{A} = (A, d_A)$. The *Hausdorff metric* is defined as follows:

$$\begin{aligned}d(\underline{A_1}, \underline{A_2}) = \quad & \max\{\sup_{x \in A_1} \inf_{y \in A_2}\{d_A(x,y)\}, \\ & \sup_{y \in A_2} \inf_{x \in A_1}\{d_A(x,y)\}\}.\end{aligned}$$

Note, in the trivial example of ordinary sets (without similarity), it is straightforward that two induced subobjects (ordinary subsets) of a given set can only be similar if they are equal, i.e., if they share all the elements.

The following theorem generalizes the above-defined, special notions of similarity between induced subobjects and proposes a similarity measure in $IndSub(\underline{A})$.

Theorem 2. Let $\underline{A_1} = (A_1, L_A, \rho_A)$ and $\underline{A_2} = (A_2, L_A, \rho_A)$ be two induced subobjects of the set with similarity $\underline{A} = (A, L_A, \rho_A)$. The Egli-Milner ordering from reflexive sets and the Hausdorff metric from bounded metric spaces can be generalized to sets with similarity as follows:

$$\begin{aligned}&\rho(\underline{A_1}, \underline{A_2}) = \\ &\quad = \Big(\bigwedge_{x \in A_1} \bigvee_{y \in A_2} \rho_A(x,y) \Big) \wedge \Big(\bigwedge_{y \in A_2} \bigvee_{x \in A_1} \rho_A(x,y) \Big),\end{aligned}$$

where all the meets and joins are computed in the complete lattice L_A.

The proof of this theorem and some highly-desirable properties of the generalized similarity measure ρ, such as

i.) the empty induced subobject is completely dissimilar to any other induced subobject and

ii.) every induced subobject is most similar to itself,

are given in (7). Note, if infinite sets are allowed, theorem 2 requires from the sets with similarity to be equipped with complete (!) lattices (see definition 1).

5 Approximate searches

As already explained, tables and answers to queries are modeled as induced subobjects. Each column is equipped with its own measure of similarity (integrated within sets with similarity), and from all these we build the measure of similarity for the whole table (see definition 4), which can be used to make comparisons between pairs of rows, find rows whose distance from some origin falls into a certain range, find nearest neighboring rows or closest pairs of rows. Hence we can perform all types of similarity search.

Moreover, the measure of similarity ρ (see theorem 2) between induced subobjects could serve to measure the nearness or exchangeability of the exact and the *relaxed answer* to a query, for comparing instances of a time-dependent table, or track changes made to a table. While in the special case of reflexive sets, the Egli-Milner relation tells us only when a table or an answer is interchangeable with another one, in the special case of bounded metric spaces, the Hausdorff metric allows a more fine-grained control of relaxation.

Example 1. Let tables 1 and 2 contain data about the users of an Internet forum at two consecutive days (day 1 and day 2).

NAME	NICK	CITY
Marko	obi	Maribor
Maja	maja	Ljubljana
Darko Koren	dare	Koper

Table 1: Table of users at day 1.

NAME	NICK	CITY
Hujs Marko	marko	Pragersko
Maja	maja	Ljubljana
Darko Koren	dare	Koper
Meta Novak	metan	Maribor
Jernej	jernej	Kranj

Table 2: Table of users at day 2.

The relational schema of tables 1 and 2 is

$$[\mathcal{P} : \text{NAME}, \mathcal{N} : \text{NICK}, \mathcal{C} : \text{CITY}],$$

where the domains corresponding to the atributes $\mathcal{P}$, $\mathcal{N}$, and $\mathcal{C}$ are the following sets with similarity:

$$\begin{aligned}\text{NAME} &= (\text{Strings}, L_{\text{Strings}}, \rho_{\text{Strings}}),\\ \text{NICK} &= (\text{Strings}, L_{\text{Strings}}, \rho_{\text{Strings}}),\\ \text{CITY} &= (\text{Cities}, L_{[0,\infty]}, d_{Cities}).\end{aligned}$$

Here, Strings is the set of all strings of maximum length 30 and Cities is the set of all possible cities, towns, and villages in the world. The complete lattice L_{Strings} is the linearly ordered set $\{0, 1, \ldots, 30\}$ with the order relation $\geq$ and $L_{[0,\infty]}$ is the complete lattice from definition 3. The measures of similarity are defined as follows:

- The measure of similarity ρ_{Strings} is the Damerau-Levenshtein distance (12), given as the minimum number of operations needed to transform one string into the other, where an operation is an insertion, deletion, or substitution of a single character or a transposition of two characters. Since the length of the strings is bounded by 30, the Damerau-Levenshtein distance is at least 0 (the greatest element in the lattice L_{Strings}) and at most 30 (the least element in the lattice L_{Strings}).

- The measure of similarity $d_{Slovenia}$ calculates the similarity of two cities as their air distance given as the Euclidean distance (in kilometres) between the Gauss-Krüger coordinates of the city centers (we have used the tool from http://www2.arnes.si/).

The measure of similarity corresponding to the Cartesian product

$$\underline{\text{USERS}} = \text{NAME} \times \text{NICK} \times \text{CITY}$$

of given sets with similarity is defined in accordance with definition 4:

$$\begin{aligned}&\rho_{\text{Users}}((p1, n1, c1), (p2, n2, c2)) =\\ &\quad = (\rho_{\text{Strings}}(p1, p2), \rho_{\text{Strings}}(n1, n2), d_{Cities}(c1, c2)),\end{aligned}$$

where $(p1, n1, c1)$ and $(p1, n1, c1)$ are two rows of the given table instance, i.e., elements of the Cartesian product of sets:

$$\text{Users} = \text{Strings} \times \text{Strings} \times \text{Cities}.$$

For instance, the similarity between the first rows of the two tables 1 and 2 is equal to $(5, 5, 18.3)$, but the similarity of the last two rows of table 2 is equal to $(9, 5, 189.9)$. Clearly, $(9, 5, 189.9) \geq (5, 5, 18.3)$, which means that the first pair of rows is more similar than the second one, i.e., the similarity value of the first pair is higher in the complete lattice

$$L_{\text{Users}} = L_{\text{Strings}} \times L_{\text{Strings}} \times L_{[0,\infty]}$$

than the similarity value of the second pair. Note, since L_{Users} is not linearly ordered, there are also uncomparable elements in the lattice.

Furthermore, the measure of similarity ρ between induced subobjects $\underline{A}$ and $\underline{B}$ of the Cartesian product $\underline{\text{USERS}}$ can be defined in accordance with theorem 2:

$$\begin{aligned}&\rho(\underline{A}, \underline{B}) =\\ &\quad = \Big(\bigwedge_{x \in A} \bigvee_{y \in B} \rho_{\text{Users}}(x, y) \Big) \wedge \Big(\bigwedge_{y \in B} \bigvee_{x \in A} \rho_{\text{Users}}(x, y) \Big),\end{aligned}$$

where all the meets and joins are computed in the complete lattice L_{Users}. Hence the similarity between the given table instances is equal to $(9, 4, 106.9)$. The similarity would certainly decrease if one of the tables would be increased in size by users living far from Slovenia and/or have or use much longer names or nicks. Clearly, if the similarity measures integrated within the sets with similarity were changed, the similarity value between the two table instances would also change and possibly have a different interpretation. Hence the usefullness of the calculated similarity values depends highly on the definitions of the basic similarity measures.

Example 2. Let table 3 contain a portion of public-transport bus routes in Ljubljana (Slovenia).

The relational schema of table 3 corresponding to relation BUSES is

$$[\mathcal{R} : \text{ROUTE}, \mathcal{D} : \text{DEPATURE}, \mathcal{DT} : \text{DTIME},\\ \mathcal{A} : \text{ARRIVAL}, \mathcal{AT} : \text{ATIME}],$$

where the domains corresponding to the atributes $\mathcal{R}$, $\mathcal{D}$, $\mathcal{DT}$, $\mathcal{A}$, and $\mathcal{AT}$ are the following sets with similarity:

$$\begin{aligned}\text{ROUTE} &= (\text{Buses}, L_2, \sigma_{\text{Buses}}),\\ \text{DEPARTURE} &= (\text{Stops}, L_{\text{Stops}}, \sigma_{\text{Stops}}),\\ \text{DTIME} &= (\text{Time}, L_{\text{Time}}, \sigma_{\text{Time}}),\\ \text{ARRIVAL} &= (\text{Stops}, L_{\text{Stops}}, \sigma_{\text{Stops}}),\\ \text{ATIME} &= (\text{Time}, L_{\text{Time}}, \sigma_{\text{Time}}).\end{aligned}$$

Here, Buses and Stops are the sets of bus routes and bus stops in Ljubljana, respectively. Time is the set of all possible times of the form HH:MM, where HH denotes hours written as $00, 01, \ldots, 23$ and MM denotes minutes written as $00, 01, \ldots, 59$. The complete lattice L_{Stops} is the linearly ordered set $\{0, 1, \ldots, M, \infty\}$ of non-negative integers (smaller than the number of all bus stops M) and the infinity value ∞ with the order relation $\geq$. The complete lattice L_{Time} is the linearly ordered set Time with 23:59 being the least element and 00:00 being the greatest element. Moreover, lattice L_2 is the lattice of boolean values from definition 2.

The measures of similarity are defined as follows:

- The measure of similarity σ_{Buses} says 1 if the given bus routes are equal and 0 if they are not.

- The measure of similarity σ_{Stops} calculates the similarity of the given bus stops as the minimum number of bus stops needed to pass by bus to come from the first bus stop to the second. If it is impossible to do this, the similarity value given is equal to ∞.

ROUTE	DEPARTURE	DTIME	ARRIVAL	ATIME	Exchangeability
9 (Štep. naselje-Trnovo)	145 (Emona)	9:58	026 (Konzorcij)	10:25	(1,0,23:58,4,00:25)
5 (Štep. naselje-Podutik)	145 (Emona)	10:10	025 (Hotel Lev)	10:26	(1,0,00:10,6,00:26)
13 (Sostro-Bežigrad)	145 (Emona)	10:11	059 (Bavarski dvor)	10:24	(1,0,00:11,5,00:24)
9 (Štep. naselje-Trnovo)	145 (Emona)	10:14	026 (Konzorcij)	10:41	(1,0,00:14,4,00:41)
6 (Črnuče-Dolgi most)	058 (Bavarski dvor)	10:29	034 (Hajdrihova)	10:32	(1,6,00:29,0,00:32)
1 (Vižmarje-Mestni log)	024 (Kolizej)	10:36	034 (Hajdrihova)	10:49	(1,7,00:36,0,00:49)
6 (Črnuče-Dolgi most)	026 (Konzorcij)	10:41	034 (Hajdrihova)	10:46	(1,8,00:41,0,00:46)
1 (Vižmarje-Mestni log)	026 (Konzorcij)	10:44	034 (Hajdrihova)	10:49	(1,8,00:44,0,00:49)
6 (Črnuče-Dolgi most)	026 (Konzorcij)	10:47	034 (Hajdrihova)	10:54	(1,8,00:47,0,00:54)

Table 3: Table of public-transport bus routes in Ljubljana. The last column contains data about the exchangeability of each row with an exact answer to the query given within example 2.

- The measure of similarity σ_{Time} calculates the similarity of two time moments as their difference (second minus first) in form of HH:MM.

Now consider the query *"It is 10 o'clock and I am at the Emona bus stop. Are there any buses to Hajdrihova? I would like to arrive as soon as possible."*, written in the language of relational algebra (4):

$$\sigma_{\mathcal{D}=\text{Emona}\wedge\mathcal{DT}=10{:}00\wedge\mathcal{A}=\text{Hajdrihova}\wedge\mathcal{AT}=10{:}00}(\text{BUSES}).$$

The last column of table 3 contains the calculated similarity or exchangeability values of the exact and the possibly relaxed answer (row). Notice that in table 3 there are no buses satisfying all the conditions given by the user but there are several buses that could be interesting for the user, such as the buses described by the second or the third row. These have a different destination, which is not a real handicap since the user could take another bus to come to Hajdrihova, i.e., bus routes 1 and 6, respectively. However, if we would like to suggest a suitable bus or a sequence of buses, we just need to calculate a θ-join of the given relation with the requirement that the arrival bus stop of the first and the departure bus stop of the second bus are (basically) the same, i.e., there are no bus stops between them, maybe one only needs to cross the street.

6 Conclusion

We have defined the mathematical structure of sets with similarity that allows us to treat the features of richly-structured data, such as order, distance, and similarity, in a theoretically sound and uniform way. The proposed measures of similarity allow us to perform all types of similarity search.

In addition, we now briefly discuss possible implementations of the resulting databases enriched with measures of similarity. Clearly, the user should be able to query approximate or cooperative data from databases without being concerned about the internal structure of data. Hence some default similarity measures should be integrated within the database. But still, the user should have the opportunity to modify the default notions of similarity if he/she is willing to do this.

However, the question that arises is how to store the defined similarity measures. When the size of the data set A is small, the evident way to store a similarity measure $\rho_A : A \times A \to L_A$ is in tabular form, i.e., as a relation

$$\rho_A \subseteq A \times A \times L_A.$$

This kind of representation quickly becomes inefficient since it requires space quadratic in the size of A. Fortunately, in most cases the similarity measure can be easily calculated so that there is no need for storing it.

There are two typical examples of similarity measures that can be computed rather than stored. First, distance-like similarities are computed from auxiliary data, such as geographic location, duration, and various other features that only require a minimal amount of additional storage.

Second, reflexive relations are often defined in terms of deduction rules, e.g., it may be known that the relation is symmetric or transitive. In such cases we only store the base cases in a database, and deduce the rest from them. This is precisely the idea behind deductive database languages, such as *Datalog*.

References

[1] P. Buneman, P. Jung, A. Ohori (1991) Using Powerdomains to Generalize Relational Databases, *Theoretical Computer Science* 9/1, Elsevier, pp. 23–55.

[2] W. W. Chu, Q. Chen (1994) A Structured Approach for Cooperative Query Answering, *IEEE Transactions on Knowledge and Data Engineering* 6/5, IEEE Computer Society, pp. 738–749.

[3] W. W. Chu, H. Jung, K. Chiang, M. Minock, G. Chow, C. Larson (1996) CoBase: A Scalable and Extensible Cooperative Information System, *Journal of Intelligent Information Systems* 6/2-3, Springer US, pp. 223–259.

[4] E. F. Codd (1970) A Relational Model of Data for Large Shared Data Banks, *Communications of the ACM* 13/6, Association for Computing Machinery, pp. 377–387.

[5] T. Gaasterland, P. Godfrey, J. Minker (1992) An Overview of Cooperative Answering, *Journal of Intelligent Information Systems* 1/2, Springer US, pp. 123–157.

[6] T. Gaasterland, P. Godfrey, J. Minker (1992) Relaxation as a Platform of Cooperative Answering, *Journal of Intelligent Information Systems* 1/3-4, Springer US, pp. 293–321.

[7] M. Hajdinjak (2006) *Knowledge Representation and Evaluation of Cooperative Spoken Dialogue Systems*, Ph.D. thesis, Faculty of Electrical Engineering, University of Ljubljana, Ljubljana, Slovenia.

[8] M. Hajdinjak, F. Mihelič (2006) The PARADISE Evaluation Framework: Issues and Findings, *Computational Linguistics* 32/2, MIT Press, pp. 263–272.

[9] G. R. Hjaltason, H. Samet (2003) Index-Driven Similarity Search in Metric Spaces, *ACM Transactions on Database Systems* 28/4, Association for Computing Machinery, pp. 517–580.

[10] A. Motro (1988) VAGUE: A User Interface to Relational Databases that Permits Vague Queries, *ACM Transactions on Office Information Systems* 6/3, Association for Computing Machinery, pp. 187–214.

[11] A. Motro (1990) FLEX: A Tolerant and Cooperative User Interface to Databases, *IEEE Transactions on Knowledge and Data Engineering* 2/2, IEEE Computer Society, pp. 231–246.

[12] G. Navarro (2001) A guided tour to approximate string matching, *ACM Computing Surveys* 33/1, Association for Computing Machinery, pp. 31–88.

[13] W. Ng (2001) An Extension of the Relational Data Model to Incorporate Ordered Domains, *ACM Transactions on Database Systems* 26/3, Association for Computing Machinery, pp. 344–383.

[14] D. E. Rutherford (1965) *Introduction to Lattice Theory*, Oliver & Boyd, Edinburgh, London.

[15] J. D. Ullman (1988) *Principles of Database and Knowledge-Base Systems, Volume I*, Computer Science Press Inc., Rockville, Maryland.

[16] J. D. Ullman (1989) *Principles of Database and Knowledge-Base Systems, Volume II: The New Technologies*, Computer Science Press, Inc., Rockville, Maryland.