Krajča P., Vychodil V.: Basic Concepts of Relational Similarity-Based Query Language RESIQL

Department of Computer Science, Palacky University, Olomouc (17. listopadu 12, CZ–77146 Olomouc, Czech Republic)

Introduction

:: Bělohlávek and Vychodil [1, 2, 3] proposed generalization of the Codd's relational database model

:: allows similarity based queries (find a car which costs about $10,000)

:: each tuple (row in the table) has a rank indicating the degree to which tuple satisfies the given query

:: solid theoretical foundations: relational algebra and calculus

Our Goal

Create a query language which

:: corresponds to the theoretical model,

:: is independent of the physical implementation,

:: allows for implicit query optimizations,

:: is suitable for database practitioners.

Data Model

Scale of Truth Degrees

:: complete residuated lattice: $\mathbf{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1 \rangle$

:: $\langle L, \wedge, \vee, 0, 1 \rangle$ is a complete lattice with 0 and 1 being the least and greatest element of L

:: $\langle L, \otimes, 1 \rangle$ is a commutative monoid

:: $\otimes$ and $\rightarrow$ satisfy so-called adjointness property: $a \otimes b \leq c$ iff $a \leq b \rightarrow c$ for each $a, b, c \in L$

:: used by domain similarities and ranks (degrees of matches)

Example

:: $\mathsf{L} = [0, 1]$

:: $a \otimes b = \max(0, a + b - 1)$

:: $a \rightarrow b = \min(1, 1 - a + b)$

[1] Belohlavek, Vychodil: Data tables with similarity relations: functional dependencies, complete rules and non-redundant bases. In: DASFAA 2006, LNCS 3882, pp. 644–658 (2006)

[2] Belohlavek, Vychodil: Query systems in similarity-based databases: logical foundations, expressive power, and completeness. In: ACM SAC 2010, pp. 1648–1655 (2010)

[3] Belohlavek, Vychodil: Codd's relational model from the point of view of fuzzy logic. J. Logic and Computation 21:851–862 (2011)

Ranked Data Table

:: ranked data tables (shortly, RDTs) are counterparts to the ordinary data tables in the original Codd's model

:: RDTs represent stored data and results of similarity-based queries where tuples are allowed to match conditions to degrees

Definitions

:: nonempty **set Y of attributes** – names of columns

:: finite subset $R \subseteq Y$ is called a **relation scheme** (heading of the table)

:: each attribute $y \in Y$ has its **domain** D_y (set of attribute's values)

:: having a scale of truth degrees, L each domain D_y can be equipped with a map $\approx_y \colon D_y \times D_y \to L$, called a **similarity**, satisfying conditions of reflexivity and symmetry:

(i) $\approx_y(u, u) = 1$ for all $u \in D_y$;

(ii) $\approx_y(u, v) = \approx_y(v, u)$ for all $u, v \in D_y$

:: $u \approx_y v$ is interpreted as a degree to which $u \in D_y$ is similar to $v \in D_y$

:: **cartesian product** of domains D_y $(y \in R)$, denoted by $\prod_{y \in R} D_y$, is a set of all maps $r \colon R \to \bigcup_{y \in R} D_y$ such that $r(y) \in D_y$ for all $y \in R$

:: each $r \in \bigcup_{y \in R} D_y$ shall be called a **tuple** on R over domains D_y $(y \in R)$

:: a **ranked data table** on R over domains D_y with similarities $\approx_y$ $(y \in R)$ is any map

$$\mathcal{D} \colon \prod_{y \in R} D_y \to L$$

such that there are at most finitely many tuples r such that $\mathcal{D}(r) > 0$.

:: the degree $\mathcal{D}(r)$ assigned to tuple r by $\mathcal{D}$ shall be called a **rank** of tuple r in $\mathcal{D}$

Remarks

:: attributes from R denote table columns, values from D_y are table entries

:: order of tuples and columns does not matter

RESIQL

:: Relational Similarity-based Query Language

:: all queries are expressions—either **scalar** or **relational** (evaluate to an RDT; constist of relational operators and variables)

:: expressions are evaluated with the `RETRIEVE` command

Restrictions

:: relational operator: $\mathfrak{D}$ `WHERE` E

:: $\mathfrak{D}$ is a relational expression evaluating to an RDT $\mathcal{D}$ and E is a condition

:: returns an RDT which for each tuple r in $\mathcal{D}$ contains the tuple r with rank

$$\mathcal{D}(r) \otimes ||E||_r$$

:: $||E||_r$ denotes the degree to which r satisfies E

:: if the rank of the tuple is 0, the tuple is omitted

Projections, Renamings, Extensions

:: relational operator: `[`$spec_1$`, ...,` $spec_n$ `FROM` $\mathfrak{D}$`]`

:: $\mathfrak{D}$ is a relational expression, each $spec_i$ is a scalar expression, optionally along with the `AS` keyword assigning a name to an attribute

:: for each tuple r computes new tuple using $spec_1$, ..., $spec_n$

:: if duplicate tuples appear, only one tuple is preserved (supremum of their ranks is used as its rank)

:: relational operator: $\mathfrak{D}$ `AS` $prefix$

:: auxiliary operator which renames all attributes of the RDT

:: adds $prefix$ and a dot to each attribute

Joins

:: relational operator: $\mathfrak{D}_1$ `CROSS JOIN` $\mathfrak{D}_2$

:: $\mathfrak{D}_1$, $\mathfrak{D}_2$ are relational expressions evaluating to RDTs $\mathcal{D}_1$, $\mathcal{D}_2$ and $\mathcal{D}_1$ and $\mathcal{D}_2$ are RDTs on disjoint relation schemes R_1 and R_2

:: result is an RDT whose relation scheme is set-theoretic union of R_1 and R_2

:: the rank of each tuple $r = r_1 r_2$ in the cross join is the result of $\mathcal{D}_1(r_1) \otimes \mathcal{D}_2(r_2)$

Examples

Pruning Operators

:: relational operators: $\mathfrak{D}$ `ABOVE` a, $\mathfrak{D}$ `TOP` n

:: $\mathfrak{D}$ is a relational expression which evaluates to $\mathcal{D}$

:: a is a scalar expression evaluating to a rank, n is a scalar expression evaluating to a positive integer

:: `ABOVE` returns an RDTs with tuples having rank $\geq a$

:: `TOP` returns an RDT containing n tuples with the highest ranks

:: remaining tuples having the same rank as tuples among those n tuples with the highest ranks are also included

Examples

:: data tables $cars$ and $customers$ used in examples

	$name$	$price$	$type$	$year$
1.00	BMW X5	12500	SUV	2004
1.00	Ford Fiesta	11560	Wagon	2011
1.00	Ford Focus	9811	Hatchback	2011
1.00	Honda Accord	10600	Wagon	2010
1.00	Hyundai i30	11699	Hatchback	2010

	$customer$	$price$	$type$
1.00	Adams	10000	Hatchback
1.00	Black	12000	SUV
0.70	Black	11000	Wagon

:: query: "Find a hatchback for about 11,500 or less"

`RETRIEVE` $cars$ `WHERE` $type \approx_{type}$ `'Hatchback'` $\otimes$ $(price \approx_{price} 11500 \vee price < 11500)$;

	name	price	type	year
1.00	Ford Focus	9811.0	Hatchback	2011
0.80	Hyundai i30	11699.0	Hatchback	2010
0.50	Honda Accord	10600.0	Wagon	2010
0.44	Ford Fiesta	11560.0	Wagon	2011

:: an example of a complex query matching available cars to customers' demands

`RETRIEVE [` $c.name$ `AS` $name$, $c.price$ `AS` $price$, $cust.customer$ `AS` $customer$, $c.price$ `-` $cust.price$ `AS` $difference$
`FROM` $cars$ `AS` c `CROSS JOIN` $customers$ `AS` $cust$
`WHERE` $((c.price \approx_{price} cust.price) \vee (c.price \leq cust.price))$ $\otimes$ $(c.type \approx_{type} cust.type)$ `] TOP` 4;

	$name$	$customer$	$price$	$difference$
1.00	Ford Focus	Adams	9811	-189
0.70	Honda Accord	Black	10600	-400
0.50	BMW X5	Black	12500	500
0.49	Ford Fiesta	Black	11560	-440
0.49	Honda Accord	Black	10600	-1400

International Center for Information and Uncertainty
SSIU 2013: International Summer School "Information and Uncertainty"
http://mcin.upol.cz/SSWIU-2013/

esf european social fund in the czech republic
EUROPEAN UNION
MINISTRY OF EDUCATION, YOUTH AND SPORTS
OP Education for Competitiveness
INVESTMENTS IN EDUCATION DEVELOPMENT