Under consideration for publication in Math. Struct. in Comp. Science

Extending Relational Algebra with Similarities

MELITA HAJDINJAK[1] and GAVIN BIERMAN[2]

[1] *Faculty of Electrical Engineering, University of Ljubljana, Slovenia*
Email: Melita.Hajdinjak@fe.uni-lj.si
[2] *Microsoft Research, Cambridge, UK*
Email: gmb@microsoft.com

Received 26 January 2011; Revised 15 November 2011

In this paper we propose various extensions to the relational model to support similarity-based querying. We build upon the $\mathcal{K}$-relation model where tuples are assigned values from an arbitrary semiring $\mathcal{K}$, and its associated positive relational algebra, $\mathrm{RA}^{+}_{\mathcal{K}}$. We consider a recently proposed extension to $\mathrm{RA}^{+}_{\mathcal{K}}$ using a monus operation on the semiring to support negative queries and show how, surprisingly, it fails for important 'fuzzy' semirings. Instead, we suggest using a negation operator. We also consider the identities satisfied by the relational algebra $\mathrm{RA}^{+}_{\mathcal{K}}$. We show that moving from a semiring to a particular form of lattice (a De Morgan frame) yields a relational algebra that satisfies all the classical (positive) relational algebra identities. We claim that realistically to support real-world similarity queries, one must move from tuple-level annotations to attribute-level annotations. We detail how our De Morgan frame-based model can be extended to support attribute-level annotations and give worked examples of similarity queries in this setting.

1. Introduction

Cooperative systems are a particular form of intelligent information systems which aim to provide more human-friendly results compared to simple direct answers from a database (Minker, 1998; Hajdinjak and Mihelič, 2006). A core component of such systems is a treatment of imprecise data. For example, a cooperative system when faced with the query "What are the names of the professors who taught category theory in the summer semester" might respond "There were no category theory courses that summer" as opposed to the equally true, but less informative, "*empty set*" response. Or, when there were some other courses with topics from category theory, it might respond "There were no category theory courses that summer, but the Advanced Algebra and Algebraic Topology courses covered some topics from category theory".

At the heart of a cooperative system is a database where the data domains come equipped with a *similarity relation*, to denote degrees of similarity rather than simply 'equal' and 'not equal'. This notion of similarity leads to an extension of the relational model where data can be *ranked*; for example, a tuple is ranked with the degree to which

it matches a query. We shall call such a database a *similarity database*. Analogous to a classical relational database, all stored data (table) in a similarity database is also ranked (initially with some maximal ranking value). Thus ranked tables represent both stored data and the results of queries. Rank-aware queries we shall refer to as *similarity queries*.

For example, the following ranked table could be the result of the query "show all courses on category theory".

CourseId	Course Name	Lecture Room	*Ranking*
M100.3	Category theory	Maths 701	1.0
M200.6	Advanced algebra	Maths Institute 11	0.6
M200.8	Algebraic topology	Statistics Lab A	0.5
CS300.7	Programming language semantics	Comp Sci 1	0.3
...	...	...	...

There have been many attempts to define extensions of the relational model to deal with similarity querying. Most utilize fuzzy logic (Zadeh, 1965) in some way to model the ranking and to provide a framework for the action of the query operators with respect to the ranking. The ranking is typically modelled by a membership function to the unit interval, $[0, 1]$ (Schmitt and Schulz, 2004; Penzo, 2005; Rosado et al., 2006; Ma, 2006), although there are generalizations where the membership function instead maps to an alebraic structure of some kind (typically poset or lattice based) (Peeva and Kyosev, 2004; Belohlavek and Vychodil, 2006). Belohlavek and Vychodil argue that the algebraic approach is a better foundation for a fuzzy relational data model, primarily because it clarifies the choices of connectives and their interpretation. As they argue, much other work appears quite *ad hoc* in comparison. We build on this by providing another advantage: it facilitates a strong connection with other work on generalizing the relational algebra.

We begin by observing that a ranked table is simply an annotated relation in the sense of Green, Karvounarakis, and Tannen; both attach a value to every tuple. Thus our starting point is their $\mathcal{K}$-relation model (Green et al., 2007). In this model tuples in a relation are annotated with a value taken from a commutative semiring, $\mathcal{K}$. A key observation of Green et al. is that the familiar positive relational algebra can be lifted naturally over $\mathcal{K}$-relations using the underlying semiring operations. The resulting relational algebra, $\mathrm{RA}^{+}_{\mathcal{K}}$, generalizes Codd's classic relational algebra (Codd, 1970), the bag algebra (Montagna and Sebastiani, 2001), the relational algebra on c-tables (Imielinski and Lipski, 1984), the probabilistic algebra on event tables (Suciu, 2008), and the provenance algebra (Cui et al., 2000; Buneman et al., 2001). One contribution of our work is to show that with relatively little work, the $\mathcal{K}$-relation model is suitable as a basis for modelling data with similarities and simple, positive similarity queries. In particular the typical similarity ranks, including the 'fuzzy' ranks, can be recast as commutative semirings (we refer to these as *similarity semirings*). Moreover, the 'fuzzy logic' appearing in similarity query languages can be clarified and explained simply and elegantly.

Green et al. only considered positive queries and left open the problem of supporting negative query operators. Recently, Geerts and Poggi addressed this problem proposing the definition of a *monus operator* on the underlying commutative semiring, which requires restricting the class of commutative semirings to so-called *m-semirings* (Geerts and Poggi, 2010). One of the surprises of our work is that the monus-based difference operator yields the wrong answer for two important (fuzzy) similarity semirings. Thus we propose a different approach to modelling negative queries in the $\mathcal{K}$-relation model.

Because of the generality of the $\mathcal{K}$-relation model, the corresponding relational algebra does not satisfy all the identities of the classical relational model. In particular, the idempotence of union and self-join do not hold. (These do not hold in the bag algebra, so we would not expect them to hold in a generalization of that algebra.) The question remains whether there is a simple, abstract algebraic characterization of generalized relational algebras that satisfies all of the classic relational algebra identities. It turns out that restricting the annotation structure from a commutative semiring, $\mathcal{K}$, to a De Morgan frame, $\mathcal{L}$, is sufficient. We show how similarity queries fall naturally into the resulting $\mathcal{L}$-relation framework.

Previous attempts to formalize similarity querying and the $\mathcal{K}$-relation model all suffer from an expressivity problem in that there is a single annotation domain. For similarity models involving fuzzy logic (Ma and Yan, 2008), most annotate tuples with a value from the real interval $[0, 1]$, and in the $\mathcal{K}$-relation model, the annotation is taken from a single commutative semiring. In other words, even though the domains have their own notion of similarity, a tuple involving more than one must collapse them all into a single rank. For simple examples this may be sufficient but we quickly run into difficulties. Returning to our example, we might be interested in the query "show all courses on category theory held near the Engineering Faculty". Clearly the similarity relation we wish to use on the lecture room would be some form of distance. Moreover, it is somewhat arbitrary to combine the similarity of course content with the similarity of location. We propose moving to a model where every attribute is annotated with a rank, rather than every tuple. (Or, equivalently, every tuple is annotated with a tuple of ranks, one per attribute, rather than a single rank.) Again we show how similarity queries fall naturally into this extension of the $\mathcal{L}$-relation model and algebra, giving a number of detailed examples.

The paper is organized as follows. In §2 we recall the definitions of $\mathcal{K}$-relations and the positive relational algebra $\mathrm{RA}^+_{\mathcal{K}}$ (Green et al., 2007), along with its extension to support negative queries, $\mathrm{RA}^+_{\mathcal{K}}(\setminus)$ (Geerts and Poggi, 2010). In §3 we make a small fix to the definition of the relational algebra $\mathrm{RA}^+_{\mathcal{K}}$ to handle selection queries dealing with similarities, and we introduce our notion of similarity measures. We consider two important (fuzzy) similarity semirings in which the difference operator from $\mathrm{RA}^+_{\mathcal{K}}(\setminus)$ yields the wrong answer. We propose in §3.6 taking a different approach using a negation operator. In §4 we consider the problem that $\mathcal{K}$-relational algebra does not support all the classical relational identities. We show how replacing semirings with De Morgan frames, a substructure of commutative semirings, leads to the $\mathcal{L}$-relation model which induces a relational algebra that satisfies all the classical (positive) identities. In §5 we move from tuple-based annotations to attribute-based annotations. We define $\mathcal{D}$-relations and an algebra on $\mathcal{D}$-relations, called *relational algebra with similarities*, or $\mathrm{RA}_{\mathcal{D}}$. We also

develop the notion of a similarity-based join. In §6 we show our model in action and consider a scenario of a database representing bus connections in a city. Finally, in §7 we conclude and consider some related work along with plans for future work.

2. Background: The $\mathcal{K}$-relation model

In this section we recall the definitions of $\mathcal{K}$-relations and the positive relational algebra $\mathrm{RA}^+_{\mathcal{K}}$, along with $\mathrm{RA}^+_{\mathcal{K}}(\setminus)$, its extension to support negative queries. The aim of the $\mathcal{K}$-relation work was to provide a generalized framework capable of capturing various forms of annotated relations. As similarity can clearly be viewed as a form of annotation we will use this as our foundation for a model of similarity-based querying.

2.1. *Positive relational algebra* $RA^+_{\mathcal{K}}$

We first assume some base domains, or *types*, commonly written as τ, which are simply sets of ground values. We use in our examples common base types such as integers and strings. We adopt the named-attribute approach, so in our model a *schema*, U, which is written $\{a_1 \colon \tau_1, \ldots, a_n \colon \tau_n\}$, is a finite map from *attribute names* a_i to their types or domains $U(a_i) = \tau_i$. We represent an *U-tuple* as a map $t = \{a_1 \colon v_1, \ldots, a_n \colon v_n\}$ from attribute names a_i to values v_i of the corresponding domain, i.e. $t(a_i) = v_i$, where $v_i \in \tau_i$ for $i = 1, \ldots, n$. We denote the set of all U-tuples by U-Tup.

Recall that a *semiring* $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$ is an algebraic structure with two binary operations (sum $\oplus$ and product $\odot$) and two distinguished elements ($\mathbf{0} \neq \mathbf{1}$) such that $(K, \oplus, \mathbf{0})$ is a commutative monoid[†] with identity element $\mathbf{0}$, $(K, \odot, \mathbf{1})$ is a monoid with identity element $\mathbf{1}$, products distribute over sums, and $\mathbf{0} \odot a = a \odot \mathbf{0} = \mathbf{0}$ for any $a \in K$ (i.e., $\mathbf{0}$ is an annihilating element). A semiring $\mathcal{K}$ is called commutative if monoid $(K, \odot, \mathbf{1})$ is commutative.

Definition 2.1 ($\mathcal{K}$-relation (Green et al., 2007)). Let $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be a commutative semiring. A $\mathcal{K}$-relation over a schema $U = \{a_1 \colon \tau_1, \ldots, a_n \colon \tau_n\}$ is a function $A \colon U\text{-Tup} \to K$ such that its support $\{t \mid A(t) \neq \mathbf{0}\}$ is finite.

Taking this extension of relations, Green et al. proposed the following natural lifting of the classical relational operators over $\mathcal{K}$-relations.

Definition 2.2 (Positive relational algebra on $\mathcal{K}$-relations (Green et al., 2007)). Let $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be a commutative semiring. The operations of the positive relational algebra on $\mathcal{K}$, denoted by $\mathrm{RA}^+_{\mathcal{K}}$, are defined as follows:

Empty relation: For any set of attributes U, there is $\emptyset_U \colon U\text{-Tup} \to K$ such that $\emptyset_U(t) \stackrel{\text{def}}{=} \mathbf{0}$ for all U-tuples t.[‡]

[†] A monoid consists of a set equipped with a binary operation that is associative and has an identity element.

[‡] As is standard, we drop the subscript on the empty relation where it can be inferred by context.

Union: If $A, B\colon U\text{-Tup} \to K$, then $A \cup B\colon U\text{-Tup} \to K$ is defined by

$$(A \cup B)(t) \stackrel{\text{def}}{=} A(t) \oplus B(t).$$

Projection: If $A\colon U\text{-Tup} \to K$ and $V \subset U$, we write $f \downarrow V$ to be the restriction of the map f to the domain V. The projection $\pi_V A\colon V\text{-Tup} \to K$ is defined by

$$(\pi_V A)(t) \stackrel{\text{def}}{=} \sum_{(t'\downarrow V)=t \text{ and } A(t')\neq \mathbf{0}} A(t').$$

Selection: If $A\colon U\text{-Tup} \to K$ and the selection predicate $\mathbf{P}$ maps each U-tuple to either $\mathbf{0}$ or $\mathbf{1}$, then $\sigma_{\mathbf{P}} A : U\text{-Tup} \to K$ is defined by

$$(\sigma_{\mathbf{P}} A)(t) \stackrel{\text{def}}{=} A(t) \odot \mathbf{P}(t).$$

Join: If $A\colon U_1\text{-Tup} \to K$ and $B\colon U_2\text{-Tup} \to K$, then $A \bowtie B$ is the $\mathcal{K}$-relation over $U_1 \cup U_2$ defined by

$$(A \bowtie B)(t) \stackrel{\text{def}}{=} A(t \downarrow U_1) \odot B(t \downarrow U_2).$$

Renaming: If $A\colon U\text{-Tup} \to K$ and $\beta\colon U \to U'$ is a bijection, then $\rho_\beta A\colon U'\text{-Tup} \to K$ is defined by

$$(\rho_\beta A)(t) \stackrel{\text{def}}{=} A(t \circ \beta).$$

Note. Note that in the case for projection, the sum is finite since A has finite support.

The power of this definition is that it generalizes a number of proposals for annotated relations and associated query algebras.

Lemma 2.1 (Example algebras on $\mathcal{K}$-relations (Green et al., 2007)).

1 The classical relational algebra with set semantics (Codd, 1970) is given by the $\mathcal{K}$-relational algebra on the boolean semiring $\mathcal{K}_{\mathbb{B}} = (\mathbb{B}, \vee, \wedge, \mathsf{false}, \mathsf{true})$.
2 The relational algebra with bag semantics (Montagna and Sebastiani, 2001; Green et al., 2007) is given by the $\mathcal{K}$-relational algebra on the semiring of counting numbers $\mathcal{K}_{\mathbb{N}} = (\mathbb{N}, +, \cdot, 0, 1)$.
3 The Fuhr-Rölleke-Zimányi probabilistic relational algebra on event tables (Suciu, 2008) is given by the $\mathcal{K}$-relational algebra on the semiring $\mathcal{K}_{\text{prob}} = (\mathcal{P}(\Omega), \cup, \cap, \emptyset, \Omega)$ where Ω is a finite set of events and $\mathcal{P}(\Omega)$ is the powerset of Ω.
4 The Imielinksi-Lipski algebra on c-tables (Imielinski and Lipski, 1984) is given by the $\mathcal{K}$-relational algebra on the semiring $\mathcal{K}_{c\text{-table}} = (\mathsf{PosBool}(X), \vee, \wedge, \mathsf{false}, \mathsf{true})$ where $\mathsf{PosBool}(X)$ is the set of all positive boolean expressions over a finite set of variables X in which any two equivalent expressions are identified.
5 The provenance algebra of polynomials with variables from X and coefficients from $\mathbb{N}$ (Cui et al., 2000; Green et al., 2007) is given by the $\mathcal{K}$-relational algebra on the provenance semiring $\mathcal{K}_{\text{prov}} = (\mathbb{N}[X], +, \cdot, 0, 1)$.

In addition, the positive relational algebra $\text{RA}^+_{\mathcal{K}}$ satisfies many of the familiar relational equalities (Ullman, 1988; Ullman, 1989).

Proposition 2.1 (Identities of $\mathcal{K}$-relations (Green et al., 2007)). The following identities hold for the positive relational algebra on $\mathcal{K}$-relations:

— union is associative, commutative, and has identity $\emptyset$;
— selection distributes over union and product;
— join is associative, commutative and distributive over union;
— projection distributes over union and join;
— selections and projections commute with each other;
— selection with Boolean predicates gives all or nothing, $\sigma_{\mathsf{false}}(A) = \emptyset$ and $\sigma_{\mathsf{true}}(A) = A$, where $\mathsf{false}(\mathsf{t}) = \mathbf{0}$ and $\mathsf{true}(\mathsf{t}) = \mathbf{1}$;
— join with an empty relation gives an empty relation, $A \bowtie \emptyset_U = \emptyset_U$ where A is a $\mathcal{K}$-relation over a schema U;
— projection of an empty relation gives an empty relation, $\pi_V(\emptyset) = \emptyset$.

It is important to note that the properties of idempotence of union, $A \cup A = A$, and self-join, $A \bowtie A = A$, are missing from this list. These properties fail for the bag semantics and provenance, so it should not be surprising that they fail to hold for the more general model. We will return to this issue in §4.

We will find it important in the following sections to consider order on commutative semirings.

Lemma 2.2 (Preorder on a semiring). Every commutative semiring $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$ supports a *preorder* defined as follows.

$$x \preceq y \text{ iff there exists a } z \in K \text{ such that } x \oplus z = y$$

2.2. *Relational algebra $RA^{+}_{\mathcal{K}}(\setminus)$*

Geerts and Poggi recently proposed extending the $\mathcal{K}$-relation model by following a standard approach for introducing a monus operator into an additive commutative monoid (Amer, 1984). First, they restricted the class of commutative semirings by requiring that every semiring additionally satisfy the following pair of conditions.

Definition 2.3 (GP-conditions (Geerts and Poggi, 2010)). A commutative semiring $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$ is said to satisfy the *GP conditions* if the following two conditions hold.

1 The preorder $x \preceq y$ on K is a *partial order*.
2 For each pair of elements $x, y \in K$, the set $\{z \in K; x \preceq y \oplus z\}$ has a smallest element. (As $\preceq$ defines a partial order, this smallest element must be unique, if it exists.)

Definition 2.4 (m-semiring (Geerts and Poggi, 2010)). Let $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be a commutative semiring that satisfies the GP conditions. For any $x, y \in K$, we define $x \ominus y$ to be the smallest element z such that $x \preceq y \oplus z$. A (commutative) semiring $\mathcal{K}$ that can be equipped with a *monus* operator $\ominus$ is called a *semiring with monus* or *m-semiring*.

Proposition 2.2 (Identities in an m-semiring (Bosbach, 1965)). The notion of an m-semiring is characterized by the properties of commutative semirings and the following

identities involving $\ominus$.

$$\begin{aligned} x \ominus x &= \mathbf{0}, \\ \mathbf{0} \ominus x &= \mathbf{0}, \\ x \oplus (y \ominus x) &= y \oplus (x \ominus y), \\ x \ominus (y \oplus z) &= (x \ominus y) \ominus z, \\ x \odot (y \ominus z) &= (x \odot y) \ominus (x \odot z). \end{aligned}$$

Geerts and Poggi identified two equationally complete classes in the variety of m-semirings, namely (1) m-semirings that are a boolean algebra (i.e., complemented distributive lattice with distinguished elements $\mathbf{0}$ and $\mathbf{1}$), for which the monus behaves like set difference, and (2) m-semirings that are the positive cone of a lattice-ordered commutative ring, for which the monus behaves like the truncated minus of the natural numbers. We will refer to the '$-$' operation as *additive inversion*. Recall that a *lattice-ordered ring* (or l-ring) is an algebraic structure $\mathcal{K} = (K, \vee, \wedge, \oplus, -, \mathbf{0}, \odot)$ such that $(K, \vee, \wedge)$ is a lattice, $(K, \oplus, -, \mathbf{0}, \odot)$ is a ring, operation $\oplus$ is order-preserving, and for $x, y \geq \mathbf{0}$ we have $x \odot y \geq \mathbf{0}$. An l-ring is commutative if the multiplication operation $\odot$ is commutative. The set of elements x for which $\mathbf{0} \leq x$ is called the *positive cone* of the l-ring.

Example 2.1 (Example m-semirings (Geerts and Poggi, 2010)).

— The boolean semiring, $\mathcal{K}_{\mathbb{B}} = (\mathbb{B}, \vee, \wedge, \mathsf{false}, \mathsf{true})$, is a boolean algebra. We have $\mathsf{false} \ominus \mathsf{false} = \mathsf{false}$, $\mathsf{false} \ominus \mathsf{true} = \mathsf{false}$, $\mathsf{true} \ominus \mathsf{false} = \mathsf{true}$, and $\mathsf{true} \ominus \mathsf{true} = \mathsf{false}$.

— The semiring of counting numbers, $\mathcal{K}_{\mathbb{N}} = (\mathbb{N}, +, \cdot, 0, 1)$, is the positive cone of the ring of integers, $\mathbb{Z}$. The monus corresponds to the truncated minus, $x \ominus y = \max\{0, x - y\}$.

— The probabilistic semiring, $\mathcal{K}_{\text{prob}} = (\mathcal{P}(\Omega), \cup, \cap, \emptyset, \Omega)$, is a boolean algebra. The monus corresponds to set difference, $X \ominus Y = X \setminus Y$.

— In the case of the semiring of c-tables, $\mathcal{K}_{c\text{-table}} = (\mathsf{PosBool}(X), \vee, \wedge, \mathsf{false}, \mathsf{true})$, the monus cannot be defined unless negated literals are added to the base set, in which case we get a boolean algebra. For any two expressions $\phi_1, \phi_2 \in \text{Bool}(X)$ we then have $\phi_1 \ominus \phi_2 = \phi_1 \wedge \neg\phi_2$, where negation $\neg$ over boolean expressions takes truth to falsity, and vice versa, and it interchanges the meet and the join operation.

— The provenance semiring, $\mathcal{K}_{\text{prov}} = (\mathbb{N}[X], +, \cdot, 0, 1)$, is the positive cone of the ring of polynomials $\mathbb{Z}[X]$. The monus of two polynomials $f[X] = \sum_{\alpha \in I} f_\alpha x^\alpha$ and $g[X] = \sum_{\alpha \in I} g_\alpha x^\alpha$, where I is a finite subset of $\mathbb{N}^n$, corresponds to the polynomial $f[X] \ominus g[X] = \sum_{\alpha \in I} (f_\alpha \dot{-} g_\alpha) x^\alpha$, where $\dot{-}$ denotes the truncated minus on $\mathbb{N}$.

Given an m-semiring, the positive relational algebra $\text{RA}^+_{\mathcal{K}}$ can be extended with the missing difference operator as follows.

Definition 2.5 (Relational algebra on $\mathcal{K}$-relations (Geerts and Poggi, 2010)). Let $\mathcal{K}$ be an m-semiring. The algebra $\text{RA}^+_{\mathcal{K}}(\setminus)$ is obtained by extending $\text{RA}^+_{\mathcal{K}}$ with the operator:

Difference If $A, B : U\text{-Tup} \to K$, then the difference $A \setminus B : U\text{-Tup} \to K$ is defined by

$$(A \setminus B)(t) \stackrel{\text{def}}{=} A(t) \ominus B(t).$$

3. Modelling similarity using the $\mathcal{K}$-relation model

As mentioned in the introduction, our intention is to model the ranking of tuples in similarity tables using the $\mathcal{K}$-valued annotation in the $\mathcal{K}$-relation model. In this section we explore in detail whether the $\mathcal{K}$-relation model and the $\mathrm{RA}^{+}_{\mathcal{K}}(\setminus)$ algebra are fit for purpose as the underlying model for similarity data and queries.

3.1. *The selection predicate*

Our first step is to make a small change to the original Green et al. model. More specifically, we observe in Definition 2.2 that the selection predicate maps U-tuples to either the zero or the unit element of the semiring. Clearly, in a similarity context we would expect the selection predicate to reflect the *degree* of membership of a particular tuple, not just the two possibilities of full membership ($\mathbf{1}$) or non-membership ($\mathbf{0}$). Thus we propose the following generalization to the original definition.

Selection: If $A\colon U\text{-Tup} \to K$ and the selection predicate

$$\mathbf{P}\colon U\text{-Tup} \to K$$

maps each U-tuple to an element of K (instead of mapping to either $\mathbf{0}$ or $\mathbf{1}$), then $\sigma_{\mathbf{P}} A\colon U\text{-Tup} \to K$ is (still) defined by

$$(\sigma_{\mathbf{P}} A)(t) = A(t) \odot \mathbf{P}(t).$$

In the rest of this paper when we refer to the relational algebra over $\mathcal{K}$-relations, we implicitly mean this generalization.

3.2. *Similarity semirings*

Tuples in similarity databases are typically ranked with either (true or false) or with some value from the real interval $[0, 1]$ reflecting the degree of membership or relevance. If we want the rank to denote the distance to the nearest ideal tuple, tuples could be ranked with values from a real interval $[0, d_{\max}]$, where 0 and $d_{\max}$ mean the zero distance and the greatest possible distance, respectively. All three are the underlying sets of commutative semirings but because of their use we shall refer to them as *similarity semirings*.

Lemma 3.1. [Common similarity semirings]

1 The boolean semiring $\mathcal{K}_{\mathbb{B}} = (\mathbb{B}, \vee, \wedge, \mathsf{false}, \mathsf{true})$ with $\mathbb{B} = \{\mathsf{true}, \mathsf{false}\}$ is a commutative semiring.
2 The *fuzzy semiring* $\mathcal{K}_{[0,1]} = ([0, 1], \max, \min, 0, 1)$ is a commutative semiring.
3 The *distance semiring* $\mathcal{K}_{[0,d_{\max}]} = ([0, d_{\max}], \min, \max, d_{\max}, 0)$ is a commutative semiring.

3.3. *Similarity measures*

As mentioned earlier, similarity databases typically assume that all data domains come equipped with a similarity relation. We call these *similarity measures* and in terms of our $\mathcal{K}$-relation setting define them as follows.

Definition 3.1 (Similarity measures). Given a type τ and a commutative semiring $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$, a *similarity measure* is a function $\rho\colon \tau \times \tau \to K$ such that ρ is reflexive, i.e. $\rho(x, x) = \mathbf{1}$.

We follow earlier work (Shenoi and Melton, 1989) and require only reflexivity of the similarity measure. This seems to be the minimal requirement, as other properties simply don't hold in general (Hajdinjak and Bauer, 2009). For example, symmetry does not hold when similarity denotes driving distance between two points in a town because of one-way streets. Another property is transitivity, but there are a number of non-transitive similarity measures, e.g. when similarity denotes likeness between two colours.

Example 3.1 (Common similarity measures). Three common examples of similarity measures are as follows.

1 An equality measure $\rho\colon \tau \times \tau \to \mathbb{B}$ where $\rho(x, y) \stackrel{\text{def}}{=} \mathsf{true}$ if x and y are equal and false otherwise.
2 A fuzzy equality measure $\rho\colon \tau \times \tau \to [0, 1]$ where $\rho(x, y)$ expresses the degree of equality of x and y; the closer x and y are to each other, the closer $\rho(x, y)$ is to 1.
3 A distance measure $\rho\colon \tau \times \tau \to [0, d_{\max}]$ where $\rho(x, y)$ is the distance from x to y.

We assume a predefined environment of similarity measures that can be used for building queries. We assume that $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$ and for every $\mathcal{K}$-relation over a schema $U = \{a_1\colon \tau_1, ..., a_n\colon \tau_n\}$ there are similarity measures $\rho_i\colon \tau_i \times \tau_i \to K, 1 \leq i \leq n$. For a given attribute a_i we write ρ_{a_i} to denote this similarity measure. Unfortunately, within a database, all similarity measures must have the same codomain, K. We shall return to this issue in §5.

Selection queries can now be classified on whether they are based on the attribute values (as is normal in non-similarity queries) or whether they use the similarity measures. (Clearly selection queries can use constant values.) The former category we refer to as primitive predicate.

Definition 3.2 (Primitive predicate). Suppose in a schema $U = \{a_1\colon \tau_1, \ldots, a_n\colon \tau_n\}$ the types of attributes a_i and a_j coincide. Then given a commutative semiring $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$, for a given binary predicate θ, the *primitive predicate* $[a_i\, \theta\, a_j]\colon U\text{-Tup} \to K$ is defined as follows.

$$[a_i\, \theta\, a_j](t) \stackrel{\text{def}}{=} \chi_{a_i \theta a_j}(t) = \begin{cases} \mathbf{1} & \text{if } t(a_i)\, \theta\, t(a_j), \\ \mathbf{0} & \text{otherwise.} \end{cases}$$

We can also define similarity predicates which select tuples based on the similarity measures.

Definition 3.3 (Similarity predicate). Suppose in a schema $U = \{a_1\colon \tau_1, \ldots, a_n\colon \tau_n\}$ the types of attributes a_i and a_j coincide. Given a commutative semiring $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$, the *similarity predicate* $[a_i\ \texttt{like}\ a_j] : U\text{-Tup} \to K$ is defined as follows.

$$[a_i\ \texttt{like}\ a_j](t) \stackrel{\text{def}}{=} \rho_{a_i}(t(a_i), t(a_j)).$$

We can also define a symmetric version as follows.

$$[a_i \sim a_j] \stackrel{\text{def}}{=} [a_i \texttt{ like } a_j] \cup [a_j \texttt{ like } a_i],$$

where union ($\cup$) of selection predicates is defined below.

Definition 3.4 (Operations on selection predicates). Given a commutative semiring $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$, we define union and intersection of two selection predicates $\mathbf{P}_1, \mathbf{P}_2 \colon U\text{-Tup} \to K$ as follows.

$$(\mathbf{P}_1 \cup \mathbf{P}_2)(t) \stackrel{\text{def}}{=} \mathbf{P}_1(t) \oplus \mathbf{P}_2(t), \text{and}$$
$$(\mathbf{P}_1 \cap \mathbf{P}_2)(t) \stackrel{\text{def}}{=} \mathbf{P}_1(t) \odot \mathbf{P}_2(t).$$

3.4. *Example: Levenshtein Similarity*

We are now in a position to consider a simple example. This deals with similarity of strings and we select values according to their distance from some given query string. Such queries and similarity measures arise naturally in the processing of web search queries, for example.

We assume a schema $U = \{\mathsf{name}\colon \mathsf{String}, \ldots\}$ and take the commutative semiring

$$\mathcal{K}_M = (\{0, 1, \ldots, M\}, \min, \max, M, 0),$$

where M is the maximum string length of the data handled by the database system. We assume that the $\mathcal{K}$-relation function $A\colon U\text{-Tup} \to K$ maps all tuples in the relation to 0 and those not in the relation to M. We are interested in queries that measure similarity of the `name` value to some given query string, `q`. We take as an example, `q` = `"Harry S. Truman"`.

We consider a predicate

$$\mathbf{P}_{\mathsf{q}} : U\text{-Tup} \to \{0, 1, \ldots, M\}$$

that maps each tuple $t = \{\mathsf{name}\colon n, \ldots\} \in U\text{-Tup}$ to the *Levenshtein distance* (Levenshtein, 1966) between the name n of length l and the query string `q` (the string `"Harry S. Truman"` of length 15). The value $\mathbf{P}_{\mathsf{q}}(t)$ is thus the minimum number of edits needed to transform the string $t(\mathsf{name})$ into the string `"Harry S. Truman"`, with the allowable edit operations being insertion, deletion, or substitution of a single character. It can take any integer value between 0 and $\max\{15, l\}$.

Suppose t_1, t_2 and t_3 are all U-tuples having `name` values of `"Harry S. Truman"`, `"Harry Truman"`, and `"H. S. Truman"`, respectively, and take the selective query $\sigma_{\mathbf{P}_{\mathsf{q}}} A$. The query answer is a $\mathcal{K}_M$-relation with

$$(\sigma_{\mathbf{P}_{\mathsf{q}}} A)(t) = \max\{A(t), \mathbf{P}_{\mathsf{q}}(t)\},$$

where:

— $\mathbf{P}_{\mathsf{q}}(t_1) = 0$, as the name strings coincide,
— $\mathbf{P}_{\mathsf{q}}(t_2) = 3$, due to the insertion of 3 characters `"S. "` into t_2,

— $\mathbf{P}_{\mathsf{q}}(t_3) = 4$, due to the substitution of the character `"."` with `"a"` and the insertion of `"rry"` into t_3.

On the other hand, in the case of a tuple $t_4 = \{$`name: "Mary Bowman"`$, \ldots\}$, we would get $\mathbf{P}_{\mathsf{q}}(t_4) = 8$, and in the case of the tuple $t_5 = \{$`name: "Elvis Presley"`$, \ldots\}$, we would get $\mathbf{P}_{\mathsf{q}}(t_5) = 14$. Smaller values mean a smaller distance to the search string `"Harry S. Truman"` and thus a greater match with the selection condition.

3.5. *Difference operators over similarity semirings*

Here we test the monus-based extension to the $\mathcal{K}$-relation framework with two semirings that are important in the fuzzy setting of similarity queries. We will see that whilst they support a monus operation in the sense of Geerts and Poggi, the induced difference operator in the relational algebra, somewhat surprisingly, does not behave as desired.

Lemma 3.2. The fuzzy semiring, $\mathcal{K}_{[0,1]} = ([0,1], \max, \min, 0, 1)$, satisfies the GP conditions.

Proof. The order relation $\preceq$ coincides with $\leq$; the two conditions hold trivially. □

It is simple to see that the monus operator can be defined directly as follows.

$$x \ominus y = \min\{z \in [0,1]; x \leq \max\{y, z\}\} = \begin{cases} 0 & \text{if } x \leq y, \\ x & \text{if } x > y. \end{cases}$$

This induces the following difference operator in the relational algebra.

$$(A \setminus B)(t) = \begin{cases} 0 & \text{if } A(t) \leq B(t), \\ A(t) & \text{if } A(t) > B(t). \end{cases}$$

Regrettably, this is not the expected definition. First, fuzzy set difference is universally defined as $\min\{A(t), 1 - B(t)\}$ (Rosado et al., 2006). Secondly, in similarity settings only totally irrelevant tuples should be annotated/ranked with 0 and excluded as a possible answer (Hajdinjak and Mihelič, 2006). In the case of the fuzzy set difference $A \setminus B$, these are exclusively those tuples t where $A(t) = 0$ or $B(t) = 1$, and certainly not where $A(t) \leq B(t)$.

Lemma 3.3. The distance semiring, $\mathcal{K}_{[0,d_{\max}]} = ([0, d_{\max}], \min, \max, d_{\max}, 0)$, satisfies the GP-conditions.

Proof. The order relation $\preceq$ coincides with $\geq$, which is a partial order relation, so condition (1) holds. Condition (2) holds since every closed subset of the distance semiring has both a smallest and a greatest element. □

The monus operator can be defined directly as follows.

$$x \ominus y = \max\{z \in [0, d_{\max}]; x \geq \min\{y, z\}\} = \begin{cases} d_{\max} & \text{if } x \geq y, \\ x & \text{if } x < y. \end{cases}$$

This induces the following difference operator in the relational algebra.

$$(A \setminus B)(t) = \begin{cases} d_{\max} & \text{if } A(t) \geq B(t), \\ A(t) & \text{if } A(t) < B(t). \end{cases}$$

Again, in the fuzzy setting, we would expect the difference operator to be defined as $\max\{A(t), d_{\max} - B(t)\}$. Moreover, this is a continuous function in contrast to the step function behaviour of the operator above resulting from the monus definition.

From the examples above we conclude that in the fuzzy setting of similarity queries the monus operation from m-semirings does not yield a proper notion of difference in the relational algebra.

3.6. *Difference via negation*

Rather than using a monus-like operator, we propose a different approach involving negation.

Definition 3.5 (Negation). Given a set L equipped with a preorder, a *negation* is an operation $\neg : L \to L$ that reverts order, $x \leq y \implies \neg y \leq \neg x$, and is involutive, $\neg\neg x = x$.

Note. It is well-known that a preordered set may have more than one negation operation. The complement operation from complemented distributive lattices (i.e., boolean algebras) is, however, unique. It is also order-reversing and involutive and thus a negation operation (Davey and Priestley, 1990). Recall that a complemented lattice is a bounded lattice in which every element x has a complement, i.e., an element x' satisfying $x \vee x' = \mathbf{1}$ and $x \wedge x' = \mathbf{0}$.

Note. For a *bounded* poset (with bounds $\mathbf{0}$ and $\mathbf{1}$), we always have $\neg\mathbf{0} = \mathbf{1}$ and $\neg\mathbf{1} = \mathbf{0}$ (Salii, 1983). Our models in §4 will satisfy this property.

Example 3.2 (Negation operators for common similarity measures). The similarity measures from Example 3.1 all support a negation operation:

— In the boolean semiring $\mathcal{K}_{\mathbb{B}} = (\mathbb{B}, \vee, \wedge, \mathsf{false}, \mathsf{true})$, negation can be defined as complementation.

$$\neg x \stackrel{\text{def}}{=} \begin{cases} \mathsf{true} & \text{if } x = \mathsf{false}, \\ \mathsf{false} & \text{if } x = \mathsf{true}. \end{cases}$$

— In the fuzzy semiring $\mathcal{K}_{[0,1]} = ([0,1], \max, \min, 0, 1)$, ordered by relation $\leq$, we can define a negation operator as $\neg x \stackrel{\text{def}}{=} 1 - x$. (In the generalized fuzzy semiring $\mathcal{K}_{[a,b]} = ([a,b], \max, \min, a, b)$, we can define $\neg x \stackrel{\text{def}}{=} a + b - x$.)

— In the distance semiring $\mathcal{K}_{[0,d_{\max}]} = ([0, d_{\max}], \min, \max, d_{\max}, 0)$, ordered by relation $\geq$, we can define a negation operator as $\neg x \stackrel{\text{def}}{=} d_{\max} - x$.

Definition 3.6 (n-semiring). A commutative n-semiring $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1}, \neg)$ is a commutative semiring $(K, \oplus, \odot, \mathbf{0}, \mathbf{1})$ equipped with a negation operator, $\neg$: $K \to K$ (with respect to the preorder on K).

Definition 3.7 (Relational algebra on $\mathcal{K}$-relations). Let $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1}, \neg)$ be a commutative n-semiring. The algebra $\mathrm{RA}^+_{\mathcal{K}}(\neg)$ is obtained by extending $\mathrm{RA}^+_{\mathcal{K}}$ with the operator:

Difference If $A, B : U\text{-Tup} \to K$, then $A \setminus B : U\text{-Tup} \to K$ is defined as:

$$(A \setminus B)(t) \stackrel{\text{def}}{=} A(t) \odot \neg B(t).$$

Example 3.3 (Relational difference over common similarity measures). Using the negation operators from Example 3.2, the difference operators in the relational algebra $\mathrm{RA}^+_{\mathcal{K}}(\neg)$ for the common similarity measures are as follows.

— In the boolean semiring $\mathcal{K}_{\mathbb{B}} = (\mathbb{B}, \vee, \wedge, \mathsf{false}, \mathsf{true})$ we get

$$(A \setminus B)(t) = \begin{cases} \mathsf{false} & \text{if } B(t) = \mathsf{true}, \\ A(t) & \text{if } B(t) = \mathsf{false}. \end{cases}$$

Observe that in this case the monus-based difference operator and the negation-based difference operator coincide.

— In the fuzzy semiring $\mathcal{K}_{[0,1]} = ([0,1], \max, \min, 0, 1)$ we get

$$(A \setminus B)(t) = \min\{A(t), 1 - B(t)\},$$

and in the generalized fuzzy semiring $\mathcal{K}_{[a,b]} = ([a,b], \max, \min, a, b)$ we get

$$(A \setminus B)(t) = \min\{A(t), a + b - B(t)\}.$$

These coincide with the fuzzy notions of difference on $[0,1]$ and $[a,b]$, respectively, mentioned in §3.5.

— In the distance semiring $\mathcal{K}_{[0,d_{\max}]} = ([0, d_{\max}], \min, \max, d_{\max}, 0)$ we get

$$(A \setminus B)(t) = \max\{A(t), d_{\max} - B(t)\}.$$

As required in §3.5, this is a continuous function of $A(t)$ and $B(t)$, and it calculates the greatest distance $d_{\max}$ only if $A(t) = d_{\max}$ or $B(t) = 0$.

As mentioned in §2.2, there are two equationally complete classes in the variety of m-semirings: m-semirings that are boolean algebras and m-semirings that are the positive cone of a lattice-ordered commutative ring.

Lemma 3.4 (Monus in boolean algebras (Amer, 1984)). Let $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1})$ be an m-semiring. If $(K, \oplus, \odot, \mathbf{0}, \mathbf{1}, ')$ is a boolean algebra where $\oplus$ is the lattice join, $\odot$ is the lattice meet, and $'$ is the complement operation, then the monus is determined as follows.

$$x \ominus y = x \odot y'.$$

Corollary 3.1 (Relational difference on $\mathcal{K}$-relations). Suppose $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1}, ')$ is a boolean algebra where $\oplus$ is the lattice join $\vee$, $\odot$ is the lattice meet $\wedge$, and $'$ is the complement operation. If we take the underlying complement operator as the negation operator, the monus-based difference operation and the negation-based difference oper-

ation induced in the algebra on $\mathcal{K}$-relations coincide,

$$A(t) \ominus B(t) = A(t) \odot B(t)' = A(t) \wedge \neg B(t).$$

The other considered semirings, the provenance semiring and the semiring of counting numbers, are each a positive cone of a lattice-ordered commutative ring.

Lemma 3.5 (Monus in lattice-ordered commutative rings (Amer, 1984)). Let $\mathcal{K}$ be the positive cone of a lattice-ordered commutative ring $(K, \vee, \wedge, \oplus, -, \mathbf{0}, \odot)$. Then the monus is determined by

$$x \ominus y = \mathbf{0} \vee (x - y)$$

where $-y$ denotes the additive inverse of y.

Unfortunately, while the provenance semiring, $\mathcal{K}_{\text{prov}} = (\mathbb{N}[X], +, \cdot, 0, 1)$, and the semiring of counting numbers, $\mathcal{K}_{\mathbb{N}} = (\mathbb{N}, +, \cdot, 0, 1)$, both contain a monus, neither contain a negation operation. Notice that in the case of an m-semiring that is a positive cone of a lattice-ordered commutative ring, even if we had taken a whole lattice-ordered commutative ring and not just its positive cone, we still have no guarantee that it contains a negation operation.

Note (Additive inversion and negation). Observe that the additive inversion operation in a lattice-ordered commutative ring may not be a negation operation. Although it is involutive, $-(-x) = x$, it may not be order-reversing, i.e., the implication $x \leq y \Rightarrow -y \leq -x$ may not hold.

Following Definition 3.7, in a commutative n-semiring we may define a difference-like operation by $x \div y \stackrel{\text{def}}{=} x \odot \neg y$. Then $\neg x = 1 \div x$.

Proposition 3.1 (Identities in a commutative n-semiring). In a commutative n-semiring, $\mathcal{K} = (K, \oplus, \odot, \mathbf{0}, \mathbf{1}, \neg)$, where $x \div y \stackrel{\text{def}}{=} x \odot \neg y$, the following additional identities involving $\div$ hold.

$$\begin{aligned} \mathbf{0} \div x &= \mathbf{0}, \\ \mathbf{1} \div (\mathbf{1} \div x) &= x, \\ x \div (\mathbf{1} \div y) &= x \odot y, \\ (x \div y) \odot y &= x \odot (y \div y). \end{aligned}$$

Notice the differences between the properties of the monus-based difference (Proposition 2.2) and the properties of the negation-based difference (Proposition 3.1). For instance, in a commutative n-semiring we do *not* have $x \div x = \mathbf{0}$ in general.

4. From semirings to lattices

In this section we return to the fact that the $\mathcal{K}$-relational algebra does not satisfy all of the classical relational algebra identities. The question we will address is whether a substructure of commutative semirings can be identified so the induced relational algebra satisfes *all* of the classical relational identities.

4.1. *De Morgan frames*

As we observed in §2.1, the $\mathcal{K}$-relational algebra does not satisfy the properties of idempotence of union and self-join. By expanding the definitions it is easy to see that this is because, in general, the sum and product operators of a semiring are not idempotent. The inverse is also true, i.e. if the sum and product operators of a semiring $\mathcal{K}$ are idempotent, then the induced $\mathcal{K}$-relational algebra has an idempotent union and self-join.

There are several different substructures of commutative semirings in which union and self-join are idempotent. However, we take this opportunity to reconsider the *order* on the underlying structure. We have noted earlier that all semirings have a preorder and some particular semirings have a partial order (e.g. m-semirings). When querying annotated relations, it is often useful to compare rankings. Clearly, for these purposes we care for something stronger than a pre- or partial order. We could insist on a linear order, but that would preclude many semiring structures of interest, including the probabilistic semiring, $\mathcal{K}_{\text{prob}}$, from earlier and other non-linearly ordered posets and lattices from more general variants of fuzzy data models (Peeva and Kyosev, 2004). Thus we impose a weaker condition but one that is still of practical use; we require that any two ranks must have a least upper bound and a greatest lower bound. In other words we restrict the commutative semiring to be a *lattice* (with the lattice join defined as sum and the lattice meet as product) (Davey and Priestley, 1990). We note immediately that lattices satisfy all the identities from Proposition 2.1 if they are bounded and distributive, and they have idempotent join (sum) and meet (product). To support relational difference, we need to consider lattices that additionally have a negation operation.

To summarize, at this point we are proposing to move from $\mathcal{K}$-relations, where $\mathcal{K}$ is a commutative semiring, to $\mathcal{L}$-relations, where $\mathcal{L}$ is a bounded, distributive lattice with a negation operation. However, whilst this would work, such structures are not very well known in the world of mathematics, where it is more natural to drop the bound requirements, and allow infinite sums. Such structures are known as De Morgan frames.

Definition 4.1 (De Morgan frame (Salii, 1983)). A *De Morgan frame*,

$$\mathcal{L} = (L, \bigvee, \wedge, \mathbf{0}, \mathbf{1}, \neg),$$

is a complete lattice $(L, \bigvee, \wedge, \mathbf{0}, \mathbf{1})$ where finite meets distribute over arbitrary joins, i.e. $x \wedge \bigvee_i y_i = \bigvee_i (x \wedge y_i)$, and $\neg\colon L \to L$ is a negation operation.

Whilst our motivations are theoretical, infinite relations have been used in the foundational of data models before: for example, they are studied in a body of work on constraint databases (Kuper et al., 2000).

Lemma 4.1. Every boolean algebra is a De Morgan frame. Moreover, the n-semirings defined in Lemma 3.1 can be extended to De Morgan Frames: $\mathcal{L}_{\mathbb{B}} = (\mathbb{B}, \vee, \wedge, \mathsf{false}, \mathsf{true}, \neg)$, $\mathcal{L}_{[0,1]} = ([0,1], \max, \min, 0, 1, \neg)$ and $\mathcal{L}_{[0,d_{\max}]} = ([0, d_{\max}], \min, \max, d_{\max}, 0, \neg)$. We refer to these as the boolean, fuzzy and distance De Morgan frames, respectively.

There are some other lattice structures with a form of negation in the literature, such as De Morgan lattices (Hutton, 1975) and fuzzy lattices (Wang, 1986). The former are

too restrictive because not all semiring structures of interest, including the fuzzy semiring, support a complementation operation. Fuzzy lattices are a completely distributive substructure of De Morgan frames, and we do not see a need for this additional property.

Proposition 4.1 (De Morgan laws (Salii, 1983)). Given a De Morgan frame $\mathcal{L} = (L, \bigvee, \wedge, \mathbf{0}, \mathbf{1}, \neg)$, the following laws hold.

$$\neg\mathbf{0} = \mathbf{1}$$
$$\neg\mathbf{1} = \mathbf{0}$$
$$\neg(x \vee y) = \neg x \wedge \neg y$$
$$\neg(x \wedge y) = \neg x \vee \neg y$$

Note. It is important to note that negation behaves differently than the familiar notion of complements in a lattice (Davey and Priestley, 1990, Chapter 7). In particular, given a De Morgan frame $\mathcal{L} = (L, \bigvee, \wedge, \mathbf{0}, \mathbf{1}, \neg)$, it is not necessarily the case that $x \vee \neg x = \mathbf{1}$ or $x \wedge \neg x = \mathbf{0}$. Indeed, these properties would preclude the fuzzy models necessary for similarity querying because the fuzzy semiring is not complemented, i.e., for $\neg x = 1 - x$ these properties do not hold.

4.2. *The $\mathcal{L}$-relation model*

Definition 4.2 ($\mathcal{L}$-relation). Let $\mathcal{L} = (L, \bigvee, \wedge, \mathbf{0}, \mathbf{1}, \neg)$ be a De Morgan frame. An *$\mathcal{L}$-relation* over a schema $U = \{a_1 : \tau_1, \ldots, a_n : \tau_n\}$ is a function $A : U\text{-Tup} \to L$.

Definition 4.3 (Relational algebra on $\mathcal{L}$-relations). Let $\mathcal{L} = (L, \bigvee, \wedge, \mathbf{0}, \mathbf{1}, \neg)$ be a De Morgan frame. The operations of the relational algebra on $\mathcal{L}$, denoted by $\mathrm{RA}_{\mathcal{L}}$, are defined as follows:

Empty relation: For any set of attributes U there is $\emptyset_U : U\text{-Tup} \to L$ such that $\emptyset(t) \stackrel{\text{def}}{=} \mathbf{0}$ for all U-tuples t.

Union: If $A, B : U\text{-Tup} \to L$ then $A \cup B : U\text{-Tup} \to L$ is defined by

$$(A \cup B)(t) \stackrel{\text{def}}{=} A(t) \vee B(t).$$

Projection: If $A : U\text{-Tup} \to L$ and $V \subset U$, the projection of A on attributes V is defined by

$$(\pi_V A)(t) \stackrel{\text{def}}{=} \bigvee_{(t' \downarrow V) = t \text{ and } A(t') \neq \mathbf{0}} A(t').$$

Selection: If $A : U\text{-Tup} \to L$ and the selection predicate $\mathbf{P} : U\text{-Tup} \to L$ maps each U-tuple to an element of $\mathcal{L}$, then $\sigma_{\mathbf{P}} A : U\text{-Tup} \to L$ is defined by

$$(\sigma_{\mathbf{P}} A)(t) \stackrel{\text{def}}{=} A(t) \wedge \mathbf{P}(t).$$

Join: If $A : U_1\text{-Tup} \to L$ and $B : U_2\text{-Tup} \to L$, then $A \bowtie B$ is the $\mathcal{L}$-relation over $U_1 \cup U_2$ defined by

$$(A \bowtie B)(t) \stackrel{\text{def}}{=} A(t \downarrow U_1) \wedge B(t \downarrow U_2).$$

Difference: If $A, B \colon U\text{-Tup} \to L$, then $A \setminus B \colon U\text{-Tup} \to L$ is defined by

$$(A \setminus B)(t) \stackrel{\text{def}}{=} A(t) \wedge \neg B(t).$$

Renaming: If $A \colon U\text{-Tup} \to L$ and $\beta \colon U \to U'$ is a bijection, then $\rho_\beta A : U'\text{-Tup} \to L$ is defined by

$$(\rho_\beta A)(t) \stackrel{\text{def}}{=} A(t \circ \beta).$$

Note. Recall that, unlike for $\mathcal{K}$-relations, we did not require $\mathcal{L}$-relations to have finite support. As De Morgan frames are *complete* lattices, we are guaranteed the existence of the join arising from the definition of projection, so requiring finite support is redundant.

As we observed earlier, the lattice supremum $\vee$ and infimum $\wedge$ operators from De Morgan frames are idempotent and as a consequence the union and self-join in $\mathrm{RA}_{\mathcal{L}}$ are idempotent, which was not the case in $\mathrm{RA}_{\mathcal{K}}$. Consequently, $\mathrm{RA}_{\mathcal{L}}$ satisfies *all* the main positive relational algebra identities which, in terms of query optimization, means that all algebraic rewrites familiar from the classical (positive) relational algebra apply to $\mathrm{RA}_{\mathcal{L}}$ without restriction.

Proposition 4.2 (Identities of $\mathcal{L}$-relations). The following identities hold for the relational algebra on $\mathcal{L}$-relations:

— union is associative, commutative, idempotent, and has identity $\emptyset$;
— selection distributes over union and difference;
— join is associative and commutative, and distributes over union;
— projection distributes over union and join;
— selections and projections commute with each other;
— difference has identity $\emptyset$ and distributes over union and intersection;
— selection with Boolean predicates gives all or nothing, $\sigma_{\mathsf{false}}(A) = \emptyset$ and $\sigma_{\mathsf{true}}(A) = A$;
— join with an empty relation gives an empty relation, $A \bowtie \emptyset_U = \emptyset_U$ where A is an $\mathcal{L}$-relation over a schema U;
— projection of an empty relation gives an empty relation, $\pi_V(\emptyset) = \emptyset$.

Proof. It is an easy exercise to see that projection distributes over union because De Morgan frames are complete. Moreover, selections and projections commute with each other because in the De Morgan frames finite meets distribute over arbitrary joins. □

Matters are a little different for the negative identities. In fuzzy relations (Rosado et al., 2006) many of the familiar laws concerning difference do not hold. For example, it is not the case that $A \setminus A = \emptyset$, and so it is not the case in general for the $\mathcal{L}$-relational algebra. By expanding the definition of difference in the case of the common fuzzy semiring $\mathcal{L} = ([0,1], \max, \min, 0, 1)$ we get

$$(A \setminus A)(t) = \min\{A(t), 1 - A(t)\}.$$

Clearly, this is only equal to 0 when $A(t) = 0$ or $A(t) = 1$. Consequently, some identities from the classical relational algebra do not hold any more.

Proposition 4.3. In $\mathrm{RA}_{\mathcal{L}}$ the following identities hold:

$$\begin{aligned} A \cap \neg A &= A \setminus A, \\ A \cup \neg A &= \neg(A \setminus A), \\ (A \setminus B) \cap B &= A \cap (B \setminus B), \\ (A \setminus B) \cup B &= (A \cup B) \cap \neg(B \setminus B), \\ A \cap B &= A \setminus (1 \setminus B), \end{aligned}$$

where $(\neg A)(t) \stackrel{\text{def}}{=} \neg A(t)$ and $(A \cap B)(t) \stackrel{\text{def}}{=} A(t) \wedge B(t)$ and $1(t) \stackrel{\text{def}}{=} \mathbf{1}$.

Proof. These all hold by simple expansion of definitions and use of the De Morgan laws given in Proposition 4.1. For example, the second identity is verified as follows.

$$(A \cup \neg A)(t) = A(t) \vee \neg A(t) = \neg(\neg A(t) \wedge A(t)) = \neg(A \setminus A)(t).$$

□

Lemma 4.2. If $\mathcal{L}$ is a boolean algebra, the identities from Proposition 4.3 are simplified as follows.

$$\begin{aligned} A \cap \neg A &= \emptyset, \\ A \cup \neg A &= 1, \\ (A \setminus B) \cap B &= \emptyset, \\ (A \setminus B) \cup B &= A \cup B, \\ A \cap B &= A \setminus (A \setminus B). \end{aligned}$$

These coincide with the identities known from classical relational algebra.

Example 4.1. Let U be a schema with an attribute $\mathsf{loc}\colon \mathsf{Loc}$, where Loc ranges over names of locations in a city, and let $\mathcal{L}_{[0,d_{\max}]} = ([0, d_{\max}], \min, \max, d_{\max}, 0, \neg)$ be the distance De Morgan frame (see Lemma 4.1), where $d_{\max}$ is the road distance between the two most extreme locations in the city. We assume also a predefined similarity measure, $\rho_{\mathsf{loc}}(x, y)$, which denotes the shortest road distance from x to y.

Let $A : U\text{-Tup} \to [0, d_{\max}]$ be a $\mathcal{L}_{[0,d_{\max}]}$-relation over schema U. We assume initially that every U-tuple t has either desirability $A(t) = 0$ or $A(t) = d_{\max}$, reflecting whether the tuple t is considered in the city centre or not. Let q denote the city center, and take the following selection predicates both mapping from U-Tup to $[0, d_{\max}]$:

$$\begin{aligned} \mathbf{P}_{\text{road}}(t) &\stackrel{\text{def}}{=} \rho_{\mathsf{loc}}(t(\mathsf{loc}), \mathsf{q}), \\ \mathbf{P}_{\text{air}}(t) &\stackrel{\text{def}}{=} \text{the air distance from } t(\mathsf{loc}) \text{ to } \mathsf{q}. \end{aligned}$$

Take the queries

$$\begin{aligned} Q_1 &\stackrel{\text{def}}{=} \sigma_{\mathbf{P}_{\text{road}}} A, \\ Q_2 &\stackrel{\text{def}}{=} (\sigma_{\mathbf{P}_{\text{air}}} A) \setminus (\sigma_{\mathbf{P}_{\text{road}}} A). \end{aligned}$$

The first query,

$$Q_1(t) = \begin{cases} \mathbf{P}_{\text{road}}(t) & \text{if } A(t) = 0 \\ d_{\max} & \text{if } A(t) = d_{\max} \end{cases}$$

is a relation in which tuples are ranked according to their road distances to the city center, $\mathbf{P}_{\text{road}}(t)$.

The second query,

$$Q_2(t) = \begin{cases} \max\{\mathbf{P}_{\text{air}}(t), d_{\max} - \mathbf{P}_{\text{road}}(t)\} & \text{if } A(t) = 0 \\ d_{\max} & \text{if } A(t) = d_{\max} \end{cases}$$

is a relation in which a tuple, t, gets a small annotation value (high desirability) whenever $t(\mathsf{loc})$ is near the city center by air, i.e., $\mathbf{P}_{\text{air}}(t)$ is near 0, and the road distance from $t(\mathsf{loc})$ to the city center is great, i.e., $\mathbf{P}_{\text{road}}(t)$ is near $d_{\max}$.

5. From tuple-based annotations to attribute-based annotations

In this section we highlight a weakness of both the $\mathcal{K}$-relation and $\mathcal{L}$-relation models. Whilst both are sufficient for very simple examples of similarity querying, there is a serious complication when considering more involved examples. This stems from the fact that there is only a *single* annotation semiring/De Morgan frame. In other words, all tuples across all the tables in the database and intermediate relations in queries must be annotated with a value from the same De Morgan frame. As we would like to support simultaneously several different similarity measures (e.g., similarity of strings, driving distance between cities, likelihood of objects to be equal), and use these different measures in our queries (even within the same query), this is clearly insufficient. We could, of course, combine all the semirings/De Morgan frames required into one semiring/De Morgan frame, but this would quickly become quite cumbersome. Moreover, it is highly non-compositional, as every query needs to involve all the similarity domains in the database, whether they are needed in the query or not. We feel that it is conceptually cleaner to move from a tuple-annotated model to an attribute-annotated model; in other words, every attribute is associated with its own De Morgan frame.

5.1. *The $\mathcal{D}$-relation model*

In this section we generalize the $\mathcal{L}$-relation model by annotating all attributes in a relation individually. There are a number of ways of setting up the technical machinery, but for ease of comparison we shall again mirror the definitions of the $\mathcal{K}$-relation and $\mathcal{L}$-relation models. We generalize an $\mathcal{L}$-relation, which is a map from a tuple to an annotation value from a De Morgan frame, to a *$\mathcal{D}$-relation*, which is a map from a tuple to a corresponding tuple containing an annotation value for every element in the source tuple (what we refer to as a De Morgan frame tuple).

Definition 5.1 (De Morgan frame schema, De Morgan frame tuple, $\mathcal{D}$-relation).

— A *De Morgan frame schema*, $\mathcal{D} = \{a_1 \colon \mathcal{L}_1, ..., a_n \colon \mathcal{L}_n\}$, maps an attribute name, a_i, to a De Morgan frame, $\mathcal{L}_i = (L_i, \bigvee_i, \wedge_i, \mathbf{0}_i, \mathbf{1}_i, \neg_i)$.
— A *De Morgan frame tuple*, $s = \{a_1 \colon l_1, ..., a_n \colon l_n\}$, maps an attribute name, a_i, to a De Morgan frame element, l_i.
— Given a De Morgan frame schema, $\mathcal{D}$, then a tuple s is said to be a *De Morgan frame tuple matching* $\mathcal{D}$ if s and $\mathcal{D}$ have the same domain, $dom(s) = dom(\mathcal{D})$.
— Given a De Morgan frame schema, $\mathcal{D}$, a schema U, then a tuple s is said to be a *De Morgan frame tuple matching* $\mathcal{D}$ *over* U if $dom(s) = dom(U) = dom(\mathcal{D})$.
— We write $\mathcal{D}(U)$-Tup to denote the set of all De Morgan frame tuples matching $\mathcal{D}$ over U.
— An $\mathcal{D}$*-relation over* U is a finite map from U-Tup to $\mathcal{D}(U)$-Tup. *Its support needs* not *be finite.*

We are now ready to define an algebra on $\mathcal{D}$-relations.

Definition 5.2 (Relational algebra with similarities).
The operations of the *relational algebra with similarities*, $\mathrm{RA}_{\mathcal{D}}$, are defined as follows:

Empty relation: For any set of attributes U and corresponding De Morgan frame schema, $\mathcal{D}$, the empty $\mathcal{D}$-relation over U, $\emptyset_U$, is defined such that $\emptyset_U(t)(a) \stackrel{\text{def}}{=} \mathbf{0}_a$ where t is a U-tuple and $\mathcal{D}(a) = (L_a, \bigvee_a, \wedge_a, \mathbf{0}_a, \mathbf{1}_a, \neg_a)$.

Union: If $A, B \colon U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$, then $A \cup B \colon U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$ is defined by

$$(A \cup B)(t)(a) \stackrel{\text{def}}{=} A(t)(a) \vee_a B(t)(a)$$

where $\mathcal{D}(a) = (L_a, \bigvee_a, \wedge_a, \mathbf{0}_a, \mathbf{1}_a, \neg_a)$.

Projection: If $A \colon U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$ and $V \subset U$, the projection of A on attributes V is defined by

$$(\pi_V A)(t)(a) \stackrel{\text{def}}{=} \bigvee_{(t' \downarrow V) = t \text{ and } A(t')(a) \neq \mathbf{0}_a} A(t')(a)$$

where $\mathcal{D}(a) = (L_a, \bigvee_a, \wedge_a, \mathbf{0}_a, \mathbf{1}_a, \neg_a)$.

Selection: If $A \colon U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$ and the selection predicate $\mathbf{P} \colon U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$ maps each U-tuple to an element of $\mathcal{D}(U)$-Tup, then $\sigma_{\mathbf{P}} A \colon U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$ is defined by

$$(\sigma_{\mathbf{P}} A)(t)(a) \stackrel{\text{def}}{=} A(t)(a) \wedge_a \mathbf{P}(t)(a)$$

where $\mathcal{D}(a) = (L_a, \bigvee_a, \wedge_a, \mathbf{0}_a, \mathbf{1}_a, \neg_a)$.

Join: Let $\mathcal{D}_1 = \{a_1 \colon \mathcal{L}_1, ..., a_n \colon \mathcal{L}_n\}$ and $\mathcal{D}_2 = \{b_1 \colon \mathcal{L'}_1, ..., b_m \colon \mathcal{L'}_m\}$ be De Morgan frame schemata. Let their union, $\mathcal{D}_1 \cup \mathcal{D}_2$, contain an attribute, $c_i \colon \mathcal{L}_i$, as soon as $c_i \colon \mathcal{L}_i$ is in $\mathcal{D}_1$ or $\mathcal{D}_2$ or both. (If there is an attribute with different corresponding De Morgan frames in $\mathcal{D}_1$ and $\mathcal{D}_2$, a renaming of attributes is needed.) If $A \colon U_1\text{-Tup} \to \mathcal{D}_1(U_1)\text{-Tup}$ and $B \colon U_2\text{-Tup} \to \mathcal{D}_2(U_2)\text{-Tup}$, then $A \bowtie B$ is the $(\mathcal{D}_1 \cup \mathcal{D}_2)$-relation over $U_1 \cup U_2$ defined as follows.

$$(A \bowtie B)(t)(a) \stackrel{\text{def}}{=} \begin{cases} A(t \downarrow U_1)(a) & \text{if } a \in U_1 - U_2 \\ B(t \downarrow U_2)(a) & \text{if } a \in U_2 - U_1 \\ A(t \downarrow U_1)(a) \wedge_a B(t \downarrow U_2)(a) & \text{if } a \in U_1 \cap U_2 \end{cases}.$$

Difference: If $A, B \colon U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$, then $A \setminus B \colon U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$ is defined by

$$(A \setminus B)(t)(a) \stackrel{\text{def}}{=} A(t)(a) \wedge_a (\neg_a B(t)(a))$$

where $\mathcal{D}(a) = (L_a, \bigvee_a, \wedge_a, \mathbf{0}_a, \mathbf{1}_a, \neg_a)$.

Renaming: If $A \colon U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$ and $\beta \colon U \to U'$ is a bijection, then $\rho_\beta A \colon U'\text{-Tup} \to \mathcal{D}(U')\text{-Tup}$ is defined by

$$(\rho_\beta A)(t)(a) \stackrel{\text{def}}{=} A(t)(\beta(a)).$$

As in the case of $\mathcal{L}$-relations we require that every tuple outside of a similarity database is ranked with the minimal De Morgan frame tuple, $\{a_1 \colon \mathbf{0}_1, \ldots, a_n \colon \mathbf{0}_n\}$, and every other tuple is ranked either with the maximal De Morgan frame tuple, $\{a_1 \colon \mathbf{1}_1, \ldots, a_n \colon \mathbf{1}_n\}$, or a smaller De Morgan frame tuple expressing a lower degree of containment of the tuple in the database.

The relational algebra with similarities, $\mathrm{RA}_{\mathcal{D}}$, satisfies the following relational algebra identities.

Proposition 5.1 (Identities of $\mathcal{D}$-relations). The following identities hold for the relational algebra on $\mathcal{D}$-relations:

— union is associative, commutative, idempotent, and has identity $\emptyset$;
— selection distributes over union and difference;
— join is associative and commutative, and distributes over union;
— projection distributes over union and join;
— selections and projections commute with each other;
— difference has identity $\emptyset$ and distributes over union and intersection;
— selection with Boolean predicates gives all or nothing, $\sigma_{\mathsf{false}}(A) = \emptyset$ and $\sigma_{\mathsf{true}}(A) = A$, where $\mathsf{false}(\mathsf{t})(\mathsf{a}) = \mathbf{0}_{\mathsf{a}}$ and $\mathsf{true}(\mathsf{t})(\mathsf{a}) = \mathbf{1}_{\mathsf{a}}$ for $\mathcal{D}(a) = (L_a, \bigvee_a, \wedge_a, \mathbf{0}_a, \mathbf{1}_a, \neg_a)$;
— join with an empty relation gives an empty relation, $A \bowtie \emptyset_U = \emptyset_U$ where A is a $\mathcal{D}$-relation over a schema U;
— projection of an empty relation gives an empty relation, $\pi_V(\emptyset) = \emptyset$.

Example 5.1. We take the schema $U = \{\mathsf{name} \colon \mathsf{String}, \mathsf{location} \colon \mathsf{String}\}$, which records pairs of customer names and their current location, and the De Morgan frame schema $\mathcal{D} = \{\mathsf{name} \colon \mathcal{L}_M, \mathsf{location} \colon \mathcal{L}_{[0,d_{\max}]}\}$ where $\mathcal{L}_M$ is the De Morgan Frame corresponding to the Levenshtein similarity semiring from Example 3.4 and the second, $\mathcal{L}_{[0,d_{\max}]}$, is the distance De Morgan frame.

Let us imagine that we have a relation A of customers and their location in which it

is anticipated that several rows, whilst different, actually denote the same information.[§] In the real-world this is common; people use variants of their name (e.g. "Bill Clinton", "W. J. Clinton") and of their city ("Manhattan", "New York"). Within the similarity setting, such data can be naturally queried. Imagine a select query $\sigma_{\mathbf{P}_{\mathsf{N},\mathsf{L}}} A$ asking for a certain customer with name N at location L, and let the selection predicate

$$\mathbf{P}_{\mathsf{N},\mathsf{L}} : U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$$

map a U-tuple $t = \{\mathsf{name}\colon n, \mathsf{location}\colon l\}$ to a De Morgan frame tuple

$$\mathbf{P}_{\mathsf{N},\mathsf{L}}(t) = \{\mathsf{name}\colon k, \mathsf{location}\colon d\}$$

where k is the Levenshtein distance between the names n and N, and d is the distance between locations l and L. Using these values, it is simple to see that the query $\sigma_{\mathbf{P}_{\mathsf{N},\mathsf{L}}} A$ satisfies the following.

$$(\sigma_{\mathbf{P}_{\mathsf{N},\mathsf{L}}} A)(t)(a) = \begin{cases} \max\{A(t)(a), k\} & \text{if } a = \mathsf{name}, \\ \max\{A(t)(a), d\} & \text{if } a = \mathsf{location}. \end{cases}$$

Then the above query succeeds in identifying the tuples that are closest to the customer with name N at location L by giving those the smallest annotation values in the associated De Morgan frame tuple.

5.2. *The selection predicate*

Each of the similarity measures associated with the attributes now maps to its own De Morgan frame. We assume that for every $\mathcal{D}$-relation over U, where $\mathcal{D} = \{a_1\colon \mathcal{L}_1, ..., a_n\colon \mathcal{L}_n\}$ and $\mathcal{L}_i = (L_i, \bigvee_i, \wedge_i, \mathbf{0}_i, \mathbf{1}_i, \neg_i)$ and $U = \{a_1\colon \tau_1, ..., a_n\colon \tau_n\}$, there is a similarity measure $\rho_i\colon \tau_i \times \tau_i \to L_i, 1 \leq i \leq n$. For a given attribute a_i we will simply write ρ_{a_i} to denote this similarity measure, rather than fold it into the definition of $\mathcal{D}$-relations.

In the $\mathcal{D}$-relation model, we redefine primitive and similarity predicates.

Definition 5.3. Suppose in a schema $U = \{a_1\colon \tau_1, \ldots, a_n\colon \tau_n\}$ the types of attributes a_i and a_j coincide. Then for a given binary predicate θ, the *primitive predicate*

$$[a_i\, \theta\, a_j]\colon U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$$

is defined as follows.

$$[a_i\, \theta\, a_j](t)(a_k) \stackrel{\text{def}}{=} \begin{cases} \chi_{a_i \theta a_j}(t) & \text{if } k = i \text{ or } k = j, \\ \mathbf{1}_k & \text{otherwise.} \end{cases}$$

In words, $[a_i\, \theta\, a_j]$ has value $\mathbf{1}$ in every attribute except a_i and a_j, where it behaves as the characteristic map of θ defined as follows.

[§] Initially all tuples in A are annotated with $\{\mathsf{name}\colon 0, \mathsf{location}\colon 0\}$, and those not in the relation with $\{\mathsf{name}\colon M, \mathsf{location}\colon d_{\max}\}$.

$$\chi_{a_i \theta a_j}(t) \stackrel{\text{def}}{=} \begin{cases} \mathbf{1}_k & \text{if } t(a_i)\,\theta\,t(a_j), \\ \mathbf{0}_k & \text{otherwise.} \end{cases}$$

Similarity predicates rank tuples based on the similarity measures.

Definition 5.4. Suppose in a schema $U = \{a_1 \colon \tau_1, \ldots, a_n \colon \tau_n\}$ the types of attributes a_i and a_j coincide. The similarity predicate $[a_i \texttt{ like } a_j] : U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$ is defined as follows.

$$[a_i \texttt{ like } a_j](t)(a_k) \stackrel{\text{def}}{=} \begin{cases} \rho_{a_i}(t(a_i), t(a_j)) & \text{if } a_k = a_i, \\ \rho_{a_j}(t(a_i), t(a_j)) & \text{if } a_k = a_j, \\ \mathbf{1}_k & \text{otherwise.} \end{cases}$$

In words, $[a_i \texttt{ like } a_j]$ measures similarity of attributes a_i and a_j, each with its own similarity measure. The symmetric version is defined as follows.

$$[a_i \sim a_j] \stackrel{\text{def}}{=} [a_i \texttt{ like } a_j] \cup [a_j \texttt{ like } a_i].$$

Now union and intersection of selection predicates are computed component-wise.

The selection criterion from Example 5.1, for instance, may be expressed by the selection predicate $[\mathsf{name} \sim a] \cap [\mathsf{location} \sim b]$, where a and b are the name and the place of living of a chosen customer, respectively.

5.3. *Additional algebraic operations*

There are some other useful operations on $\mathcal{D}$-relations that can be expressed in terms of the basic operations.

Definition 5.5 (Intersection). We define intersection $A \cap B : \mathcal{D}\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$ of relations $A, B : U\text{-Tup} \to \mathcal{D}(U)\text{-Tup}$ as follows.

$$(A \cap B)(t)(a) \stackrel{\text{def}}{=} A(t)(a) \wedge_a B(t)(a)$$

where $\mathcal{D}(a) = (L_a, \bigvee_a, \wedge_a, \mathbf{0}_a, \mathbf{1}_a, \neg_a)$.

Proposition 5.2. Intersection $A \cap B$ is equivalent to selection $\sigma_{\mathbf{P}} A$ with an appropriately chosen selection predicate. That is, if $\mathbf{P}(t) \stackrel{\text{def}}{=} B(t)$, for all $a \in U$ we have

$$(A \cap B)(t)(a) = (\sigma_{\mathbf{P}} A)(t)(a).$$

Proof. Clearly, if $\mathbf{P}(t) \stackrel{\text{def}}{=} B(t)$, we have $(\sigma_{\mathbf{P}} A)(t)(a) = A(t)(a) \wedge_a B(t)(a)$. □

Definition 5.6 (Product). We define product of $A : U_1 \to \mathcal{D}_1(U_1)$ and $B : U_2 \to \mathcal{D}_2(U_2)$ with $U_1 \cap U_2 = \emptyset$ as a $(\mathcal{D}_1 \cup \mathcal{D}_2)$-relation over $U_1 \cup U_2$ as follows.

$$(A \times B)(t) \stackrel{\text{def}}{=} A(t \downarrow U_1) \cup B(t \downarrow U_2).$$

If the joining relations have no common attributes, product and join coincide.

Proposition 5.3. If $A : U_1 \to \mathcal{D}_1(U_1)$ and $B : U_2 \to \mathcal{D}_2(U_2)$ and $U_1 \cap U_2 = \emptyset$, product is equivalent to join:

$$(A \times B)(t)(a) = (A \bowtie B)(t)(a).$$

Proof. For the proof observe

$$(A \times B)(t)(a) = \begin{cases} A(t \downarrow U_1)(a) & \text{if } a \in U_1 \\ B(t \downarrow U_2)(a) & \text{if } a \in U_2 \end{cases}.$$

□

5.4. *Similarity-based joins*

Given the similarity measures associated with attributes, it is possible to define similarity-based variants of other familiar relational operators. In this section we develop the notion of a similarity-based join. The intuition of this operator is clear: we join two rows not only when their join-attributes have equal associated values, but when the values are similar.

Definition 5.7 (Similarity-based joins). Consider a schema U, containing attribute $a\colon \tau$, and a schema V, containing attribute $b\colon \tau$. Note the types of the attributes a and b must be the same, but their supporting similarity measures ρ_a and ρ_b need not. We *join* $\mathcal{D}$-relations A (over U) and B (over V) with respect to attributes a and b in different ways:

Equality join:

$$A \bowtie_{a=b} B \stackrel{\text{def}}{=} \sigma_{[a=b]}(A \times B),$$

Similarity join:

$$A \bowtie_{a \texttt{ like } b} B \stackrel{\text{def}}{=} \sigma_{[a \texttt{ like } b]}(A \times B),$$

Symmetric similarity join:

$$A \bowtie_{a \sim b} B \stackrel{\text{def}}{=} \sigma_{[a \sim b]}(A \times B).$$

For simplicity we considered only the case of a binary join; the generalization to a join of several pairs of attributes is straightforward—the selection predicate is defined as the intersection ($\cap$) of several primitive and/or similarity predicates. Similar notions of similarity-based joins have been developed (only) on a fixed structure of truth degrees for all attributes (Adali et al., 1998; Penzo, 2005; Belohlavek and Vychodil, 2006).

Since $=$ and $\sim$ are symmetric, we may commute the compared attributes:

Proposition 5.4. We have:

$$A \bowtie_{a=b} B = A \bowtie_{b=a} B,$$
$$A \bowtie_{a \sim b} B = A \bowtie_{b \sim a} B.$$

This, of course, does not hold for the (ordinary) similarity joins since $[a \texttt{ like } b]$ and $[b \texttt{ like } a]$ are in general different similarity predicates.

6. Extended example

We consider a simplified scenario of a database representing bus connections in a city. A cooperative database system should consider several connecting rides as well as possible routes that include (small!) sections of walking between stations. Clearly such a system requires knowledge of the distance between arbitrary locations in a city. Moreover, a notion of similarity between arrival and departure times allows us to consider interconnection timings.

Consider the relational schema

$$U = \{\, \mathsf{from}\colon \mathsf{Loc}, \mathsf{to}\colon \mathsf{Loc}, \mathsf{dep}\colon \mathsf{Time}, \mathsf{arr}\colon \mathsf{Time}, \mathsf{bus}\colon \mathsf{Bus\_id} \,\},$$

where Loc, Time, and Bus_id range over names of bus stations, bus departure/arrival times in minutes, and bus ids, respectively. Assume also the De Morgan frame schema

$$\mathcal{D} = \{\, \mathsf{from}\colon \mathcal{L}_{[0,d_{\max}]}, \mathsf{to}\colon \mathcal{L}_{[0,d_{\max}]}, \mathsf{dep}\colon \mathcal{L}_{[0,t_{\max}]}, \mathsf{arr}\colon \mathcal{L}_{[0,t_{\max}]}, \mathsf{bus}\colon \mathcal{L}_{\mathbb{B}} \,\},$$

where $d_{\max}$ is the walking time distance between two furthest locations in the city and $t_{\max} = 24 \cdot 60 = 1440$ is the number of minutes in a day, and a predefined environment of similarity measures:

$$\begin{aligned}
\rho_{\mathsf{from}}(x,y) = \rho_{\mathsf{to}}(x,y) &\stackrel{\text{def}}{=} \text{time needed to walk from } x \text{ to } y,\\
\rho_{\mathsf{dep}}(x,y) &\stackrel{\text{def}}{=} \begin{cases} x - y & \text{if } x \geq y,\\ t_{\max} & \text{otherwise,} \end{cases}\\
\rho_{\mathsf{arr}}(x,y) &\stackrel{\text{def}}{=} \rho_{\mathsf{dep}}(y,x),\\
\rho_{\mathsf{bus}}(x,y) &\stackrel{\text{def}}{=} \mathsf{true}.
\end{aligned}$$

The similarity $\rho_{\mathsf{dep}}(x,y)$ between the bus departure time, x, and the time when we arrive to the station, y, denotes how long we have to wait for the bus to depart. If we come too late, we miss the bus. Similarly, the similarity $\rho_{\mathsf{arr}}(x,y)$ is defined as the penalty when we arrive at time x and would like to arrive at time y, where coming too late is deemed catastrophic. Because we (usually) don't care which bus we take, all bus ids are declared similar by ρ_{bus}.

Let buses be a bus timetable relation, a $\mathcal{D}$-relation over schema U. Prior to any querying, we assume that each timetable entry, tuple t, has desirability

$$1 = \{\mathsf{from}\colon 0, \mathsf{to}\colon 0, \mathsf{dep}\colon 0, \mathsf{arr}\colon 0, \mathsf{bus}\colon \mathsf{true}\},$$

and all other tuples have desirability

$$0 = \{\mathsf{from}\colon d_{\max}, \mathsf{to}\colon d_{\max}, \mathsf{dep}\colon t_{\max}, \mathsf{arr}\colon t_{\max}, \mathsf{bus}\colon \mathsf{false}\}.$$

We develop several ways in which the question

"How can I get from station a to station b now?"

can be expressed as a query, assuming that the current time is given by a constant now.

1 We first consider a selective query,

$$Q \stackrel{\text{def}}{=} \sigma_{\mathbf{P}}\mathsf{buses},$$

and vary the selection predicate, $\mathbf{P}$, as follows.

— If we define

$$\mathbf{P} \stackrel{\text{def}}{=} [\mathsf{from} = a] \cap [\mathsf{to} = b] \cap [\mathsf{dep} = \mathsf{now}],$$

the query Q corresponds exactly to the text of the question. It is unlikely to succeed because it asks for a bus which departs at this very moment.

— We relax the departure time by replacing equality $=$ with the similarity predicate `like` :

$$\mathbf{P} \stackrel{\text{def}}{=} [\mathsf{from} = a] \cap [\mathsf{to} = b] \cap [\mathsf{dep}\ \texttt{like}\ \mathsf{now}].$$

The answer to the resulting query consists of rows t together with their desirabilities $Q(t)$, which are De Morgan frame tuples of the form

$$Q(t) = \{\mathsf{from}\colon x, \mathsf{to}\colon y, \mathsf{dep}\colon z, \mathsf{arr}\colon 0, \mathsf{bus}\colon \mathsf{true}\}.$$

The desirability components $Q(t)(\mathsf{from}) = x$ and $Q(t)(\mathsf{to}) = y$ are either 0 or $d_{\max}$, depending on whether $t(\mathsf{from}) = a$ and $t(\mathsf{to}) = b$. The number $Q(t)(\mathsf{dep}) = z$ conveys how long we have to wait for the bus to depart. The attributes arr and bus receive the greatest value because they do not appear in the selection condition.

— If we are willing to walk from a to a different starting station, but we want to arrive exactly at b, we can use the selection predicate

$$\mathbf{P} \stackrel{\text{def}}{=} [\mathsf{from}\ \texttt{like}\ a] \cap [\mathsf{to} = b] \cap [\mathsf{dep}\ \texttt{like}\ \mathsf{now}].$$

In this case each answer t is annotated with a De Morgan frame tuple

$$Q(t) = \{\mathsf{from}\colon x, \mathsf{to}\colon y, \mathsf{dep}\colon z, \mathsf{arr}\colon 0, \mathsf{bus}\colon \mathsf{true}\},$$

where y and z are as before, whereas $Q(t)(\mathsf{from}) = x$ tells us how long it takes to walk from a to the departure station. It may happen that the bus will depart from $\mathsf{from}(t)$ before we get there.

— Further, we can support the case where the starting and ending locations, a and b, are not necessarily locations of bus stations. The selection predicate

$$\mathbf{P} \stackrel{\text{def}}{=} [\mathsf{from}\ \texttt{like}\ a] \cap [\mathsf{to}\ \texttt{like}\ b] \cap [\mathsf{dep}\ \texttt{like}\ \mathsf{now}]$$

gives useful answers for arbitrary locations a and b. The desirability $Q(t)$ of a row t contains information about the time needed to get from location a to the departure station and from the arrival station to the final destination b.

2 Now suppose we are interested in indirect connections between a and b, and let buses_1 and buses_2 be two instances of the $\mathcal{D}$-relation buses. The query

$$Q \stackrel{\text{def}}{=} \sigma_{\mathbf{P}}(\mathsf{buses}_1 \bowtie_{\mathsf{to}_1 \sim \mathsf{from}_2} \mathsf{buses}_2)$$

with the selection predicate being defined as

$$\mathbf{P} \stackrel{\text{def}}{=} [\mathsf{from}_1\ \texttt{like}\ a] \cap [\mathsf{to}_2\ \texttt{like}\ b] \cap [\mathsf{dep}_1\ \texttt{like}\ \mathsf{now}]$$

computes connecting rides from stations near a to stations near b. The schema of relation Q is $U_1 \cup U_2$, which is obtained (after a renaming of attributes) as the union

of two instances of schema U. Its De Morgan frame schema, $\mathcal{D}_1 \cup \mathcal{D}_2$, is obtained similarly. In this case each answer tuple, $t \in (U_1 \cup U_2)$-Tup, containes information about a pair of rides. It is annotated with a De Morgan frame tuple,

$$Q(t) = \{\mathsf{from}_1 : x, \mathsf{to}_1 : y, \mathsf{dep}_1 : u, \mathsf{arr}_1 : 0, \mathsf{bus}_1 : \mathsf{true},$$
$$\mathsf{from}_2 : y, \mathsf{to}_2 : z, \mathsf{dep}_2 : 0, \mathsf{arr}_2 : 0, \mathsf{bus}_2 : \mathsf{true}\}.$$

The main components of desirability $Q(t)$ are:

— x is walking time from a to the departure station,
— y is walking time between the connecting stations $t(\mathsf{to}_1)$ and $t(\mathsf{from}_2)$,
— z is walking time from the arrival station to b,
— u is waiting time until the first bus departs.

3 Lastly, if we only want to know which buses drive from station a to station b, we can simply combine a projection and a selection:

$$Q \stackrel{\text{def}}{=} \pi_{\mathsf{bus}}(\sigma_{[\mathsf{from}=a]\cap[\mathsf{to}=b]}\mathsf{buses}).$$

The answers now contain only the attribute bus, and the desirability $Q(t)$ has value true for buses that pass station a before they pass station b and false otherwise.

7. Related work and conclusions

In this paper we have considered the problem of defining a formal model suitable for modelling similarity querying. After observing that ranked tables are simply annotated relations in the sense of Green et al. we have considered the suitability of their $\mathcal{K}$-relation model for encoding similarity information and similarity queries. We have also considered a recent extension of this model to handle negative queries, due to Geerts and Poggi. We have shown that when considering the fuzzy commutative semirings that are important in the similarity setting, their extension does not behave well. We have shown that using a negation operation, rather than a monus operation, offers a solution that works in the similarity settings. We then moved on to consider the identities induced in the relational algebra by the underlying commutative semiring. To handle all the familiar identities from the relational algebra, we propose moving from commutative semirings with negation to De Morgan frames. Our final contribution is to propose moving from row-level annotations from a single semiring/De Morgan frame to attribute-level annotations, where each attribute supports its own annotation domain.

There have been a huge number of extensions to the relational model to support imprecise/uncertain data. Much of that work models probabilistic uncertainty of data, which is quite different from the kind of deterministic, fuzzy uncertainty we are interested in. The field of 'fuzzy' extensions to the relational model is also huge—Ma and Yan give a comprehensive overview (Ma and Yan, 2008). Rather than simply repeating this overview, we shall mention two closely related works. Belohlávek and Vychodil's approach to modelling similarity querying is similar to ours; taking an algebraic approach and proposing complete lattices to encode the degree of truth (Belohlavek and Vychodil, 2006). There are some small differences of detail (for example, they require similarity

measures to be symmetric), but essentially there are three major differences between our work: first, we relate our model with existing other work on annotated relations. Secondly, we consider negation whereas they focus only on positive queries, and thirdly we propose attribute-level ranking, whereas Belohlávek and Vychodil rank all tuples in the database using a single, fixed lattice structure. As we have argued, we find this to be extremely limiting in practice.

Schmitt and Schulz's work is also close to ours (Schmitt and Schulz, 2004). They define a similarity relational calculus, which combines the handling of imprecise truth values together with a traditional relational domain calculus, and show how to map the similarity relational calculus to a similarity algebra. Putting aside their use of a calculus rather than an algebra, the major difference between our work is that they consider a single, concrete rank domain (the interval $[0, 1]$) whereas we take an abstract, algebraic approach. This enables us to derive properties that *all* models must satisfy. They define a tuple rank based on a specific aggregation of the specific truth values (with similarities all taken from the unit interval) whereas we rank each tuple with a tuple of similarity values (taken from possibly different similarity domains). They do not develop important features such as similarity joins and do not attempt any comparison with the classical relational identities.

Although there are many advantages to our proposal, there are some disadvantages. First, it is clear that asking all attributes to be annotated requires more storage than simple row-level annotation. Another problem is that even if each De Morgan frame used in a De Morgan frame schema is linearly ordered, it is not the case that there is a linear order on the De Morgan frame tuples. Hence, it may not be possible to list query answers in a (decreasing) order of relevance. But this simply reflects a fact about ordered structures as opposed to any flaw in our model.

However, an ordering with decreasing relevance is guaranteed when the product of the De Morgan frames from the De Morgan frame schema is *graded* (Stanley, 1997). A *graded* or *ranked poset* is a partially ordered set P equipped with a monotone rank function $\rho : P \to \mathbb{Z}$ compatible with the ordering (so $\rho(x) < \rho(y)$ whenever $x < y$) such that whenever y covers x, then $\rho(y) = \rho(x) + 1$. Examples of graded posets are the natural numbers with the usual order, the Cartesian product of two or more sets of natural numbers with the product order being the sum of the coefficients, and the boolean lattice of finite subsets of a set ordered according to the number of elements in the subset. On the other hand, neither the integers (with the usual order) nor any interval (with more than one element) of rational or real numbers is a graded poset. Hence we cannot expect to have a rank function when querying similarity information in the $\mathcal{D}$-relation model. The De Morgan frame tuples in the answer space may be pairwise incomparable, and there may not exist a "most" desirable tuple. However, most relevant tuples (that should be offered to the user first) are those not dominated by any other tuple on all dimensions of desirability. Such tuples build the so-called *skyline* (Papadias et al., 2005). The final decision about the best choice from all the skyline tuples should be left to the user, it cannot be a part of the database system. If, nevertheless, a top-1 object (tuple) is chosen with any approximate top-k method (in favor of performance),

based on any monotone ranking function, it is known that it must be one of the skyline objects (Ilyas et al., 2008).

As we have already mentioned, our relational algebra with similarities does not generalize the provenance algebra because the provenance semiring does not have a negation operation. This is the only reason; our proposal to move to attribute-based annotations is still applicable and provides a more fine-grained provenance model. We leave as interesting future work to develop a notion of "similarity provenance" which captures the similarity between a query answer and the source tuples that have influence on it.

Acknowledgements

We should like to thank Andrej Bauer, Marcelo Fiore and especially the anonymous referee for many helpful comments and suggestions.

References

Adali, S., Bonatti, P., Sapino, M. L., and Subrahmanian, V. S. (1998). A multi-similarity algebra. *ACM*, 27:402–413.

Amer, K. (1984). Equationally complete classes of commutative monoids with monus. *Algebra Universalis*, 18(1):129–131.

Belohlavek, R. and Vychodil, V. (2006). Relational model of data over domains with similarities: An extension for similarity queries and knowledge extraction. In *Proceedings of IEEE International Conference on Information Reuse and Integration*, pages 207–213.

Bosbach, B. (1965). Komplementäre halbgruppen: Ein beitrag zur instruktiven idealtheorie kommutativer halbgruppen. *Mathematische Annalen*, 161(4):279–295.

Buneman, P., Khanna, S., and Tan, W. (2001). Why and where: A characterization of data provenance. In *Proceedings of the International Conference on Database Theory*, pages 316–330.

Codd, E. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387.

Cui, Y., Widom, J., and Wiener, J. (2000). Tracing the lineage of view data in a warehousing environment. *ACM Transactions on Database Systems*, 25:179–227.

Davey, B. and Priestley, H. (1990). *Introduction to Lattices and Order*. Cambridge University Press.

Geerts, F. and Poggi, A. (2010). On database query languages for k-relations. *Journal of Applied Logic*, 8:173–185.

Green, T., Karvounarakis, G., and Tannen, V. (2007). Provenance semirings. In *Proceedings of the Symposium on Principles of Database Systems*, pages 31–40.

Hajdinjak, M. and Bauer, A. (2009). Similarity measures for relational databases. *Informatica*, 33(2):135–141.

Hajdinjak, M. and Mihelič, F. (2006). The PARADISE evaluation framework: Issues and findings. *Computational Linguistics*, 32(2):263–272.

Hutton, B. (1975). Normality in fuzzy topological spaces. *Journal of Mathematical Analysis and Applications*, 50:74–79.

Ilyas, I., Beskales, G., and Soliman, M. (2008). A survey of top-k query processing techniques in relational database systems. *ACM Computing Surveys*, 40(4).

Imielinski, T. and Lipski, W. (1984). Incomplete information in relational databases. *Journal of the ACM*, 31(4).

Kuper, G., Libkin, L., and Paredaens, J. (2000). *Constraint Databases*. Springer Verlag.

Levenshtein, V. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10:707–710.

Ma, Z. (2006). Fuzzy database modeling of imprecise and uncertain engineering information. *Studies in Fuzziness and Soft Computing*, 195:137–158.

Ma, Z. and Yan, L. (2008). A literature overview of fuzzy database models. *Journal of Information Science and Engineering*, 24:189–202.

Minker, J. (1998). An overview of cooperative answering in databases. In *Proceedings of Conference on Flexible Query Answering Systems*, pages 282–285.

Montagna, F. and Sebastiani, V. (2001). Equational fragments of systems for arithmetic. *Algebra Universalis*, 46(3):417–441.

Papadias, D., Tao, Y., Fu, G., and Seeger, B. (2005). Progressive skyline computation in database systems. *ACM Transactions on Database Systems*, 30(1):41–82.

Peeva, K. and Kyosev, Y. (2004). *Fuzzy Relational Calculus: Theory, Applications And Softwares*, volume 22 of *Advances in Fuzzy Systems Applications and Theory*. World Scientific Publishing Company.

Penzo, W. (2005). Rewriting rules to permeate complex similarity and fuzzy queries within a relational database system. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):255–270.

Rosado, A., Ribeiro, R. A., Zadrozny, S., and Kacprzyk, J. (2006). Flexible query languages for relational databases: An overview. In Bordogna, G. and Psaila, G., editors, *Flexible databases supporting imprecision and uncertainty*, pages 3–53. Springer Verlag.

Salii, V. (1983). Quasi-boolean lattices and associations. In *Proceedings of Colloq. Math. Soc. János Bolyai*, volume 43 of *Lectures in Univ. Algebra*, pages 429–454.

Schmitt, I. and Schulz, N. (2004). Similarity relational calculus and its reduction to a similarity algebra. In *Proceedings of Symposium on Foundations of Information and Knowledge Systems*, pages 252–272.

Shenoi, S. and Melton, A. (1989). Proximity relations in the fuzzy relational database model. *Fuzzy Sets and Systems*, 31(3):285–296.

Stanley, R. (1997). *Enumerative Combinatorics*, volume 1 of *Cambridge Studies in Advanced Mathematics 49*. Cambridge University Press.

Suciu, D. (2008). Probabilistic databases. *SIGACT News*, 39(2):111–124.

Ullman, J. (1988). *Principles of Database and Knowledge-Base Systems*, volume 1. Computer Science Press.

Ullman, J. (1989). *Principles of Database and Knowledge-Base Systems*, volume 2. Computer Science Press.

Wang, G. (1986). On the structure of fuzzy lattices. *Acta Mathematica*, 29:539–543.

Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8:338–353.