On Database Query Languages for K-relations

Floris Geerts[a], Antonella Poggi[b]

[a] *University of Edinburgh*
[b] *Sapienza Università di Roma*

Abstract

The relational model has recently been extended to so-called $\mathcal{K}$-relations in which tuples are assigned a unique value in a semiring $\mathcal{K}$. A query language, denoted by $\mathcal{RA}^+_{\mathcal{K}}$, similar to the classical positive relational algebra, allows for the querying of $\mathcal{K}$-relations. In this paper, we define more expressive query languages for $\mathcal{K}$-relations that extend $\mathcal{RA}^+_{\mathcal{K}}$ with the *difference* and *constant annotations* operations on annotated tuples. These operations are natural extensions of the difference and duplicate elimination operations in the relational algebra on sets and bags, respectively. We investigate conditions on semirings under which these operations can be added to $\mathcal{RA}^+_{\mathcal{K}}$ in a natural way, and establish basic properties of the resulting query languages. Moreover, we show how the provenance semiring of Green et al. can be extended to record provenance of data in the presence of difference and constant annotations. Finally, we investigate the *completeness* of $\mathcal{RA}^+_{\mathcal{K}}$ and extensions thereof in the sense of Bancilhon and Paredaens (BP).

1. Introduction

Annotated relations appear in various contexts in the database literature. The querying of such relations involves the generalization of the relational algebra to perform corresponding operations on the annotations. Recently, a general data model (referred to as $\mathcal{K}$-relations) has been proposed for annotated relations in which tuples in a relation are assigned a unique value coming from a *semiring* $\mathcal{K}$ [12]. By varying the semiring $\mathcal{K}$, $\mathcal{K}$-relations can model the standard relational model with both set [1] and bag semantics [16], incomplete databases (positive Boolean c-tables to be more precise) [13, 15] and probabilistic databases [10, 19]. Moreover, operations that queries in the relational algebra perform on tuples can be naturally extended to operations on annotated tuples. More specifically, operations on tuples naturally translate into the algebraic operations (sum and product) in semirings. This leads to the definition of the positive relational algebra on $\mathcal{K}$-relations, or $\mathcal{RA}^+_{\mathcal{K}}$ for short [12].

The generality of semirings further allows for the definition of new data models which are of particular interest for the study of provenance of data [6, 12]. A notable example is the *provenance semiring* that allows to record provenance information of data obtained as result of positive relational algebra queries. A crucial property of this semiring, named *factorization property*, is that it is the most general semiring. That is, for any semiring $\mathcal{K}$, to evaluate queries in $\mathcal{RA}^+_{\mathcal{K}}$ on $\mathcal{K}$-relations it is sufficient to know how to evaluate these queries on the provenance semiring.

In this paper, we study query languages for $\mathcal{K}$-relations. Indeed, while some basic properties of $\mathcal{RA}^+_{\mathcal{K}}$ are already established in [12], less is known about its expressive power. Furthermore, it was left open in [12] how to incorporate difference in $\mathcal{RA}^+_{\mathcal{K}}$ to get a full relational algebra on $\mathcal{K}$-relations. Hence, our goal is twofold. On one hand, we define more expressive query languages for $\mathcal{K}$-relations that extend $\mathcal{RA}^+_{\mathcal{K}}$ with operations on annotated tuples that are natural extensions of corresponding operations of the relational

algebra. On the other hand, we investigate the expressive power of $\mathcal{RA}^+_{\mathcal{K}}$ and extensions thereof. In particular, we investigate the *completeness* of these query languages. Recall that Codd qualified a query language on relational databases as *complete* if its expressive power is at least that of the relational calculus [8]. Bancilhon [4] and Paredaens [18] independently provided a *language-independent* characterization of completeness. This characterization, known as BP-completeness, can be stated as follows: a relation R_2 is the result of a relational algebra query applied to a database R_1 if and only if (i) the active domain of R_2 is included in the active domain of R_1; and (ii) every automorphism of R_1 is also an automorphism of R_2.

The contributions of the paper can be summarized as follows:

- First, we define the query languages $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$, $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ and $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$, obtained by extending $\mathcal{RA}^+_{\mathcal{K}}$ with *difference*, *constant annotations*, and with both *difference* and *constant annotations*, respectively. Here, constant annotations correspond to a family of operators that assign annotations to tuples among a finite set of elements of the semiring, that are the *semiring generators*. Note, in particular, that extending $\mathcal{RA}^+_{\mathcal{K}}$ with these operators forces to restrict the class of semirings under consideration. Specifically, on one hand, adding difference requires the definition of a *monus* operator on the underlying semiring, which might not always be possible. We call *m-semirings* the class of semirings admitting a monus operator. On the other hand, constant annotations require the underlying semiring to be *finitely generated*, *i.e.,* to have a finite set of semiring generators. Interestingly, we observe that most semirings encountered in the literature are indeed finitely generated m-semirings.

- Second, we show how to extend the provenance semiring of [12], so that it can be used to record the provenance of data obtained as result of queries in $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$, $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ and $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$. We show that the extended provenance semirings also satisfy the factorization property.

- Finally, we naturally extend the notion of BP-completeness to the setting of $\mathcal{K}$-relations and investigate whether query languages on $\mathcal{K}$-relations proposed so far are BP-complete. In particular, we show that none of the languages $\mathcal{RA}^+_{\mathcal{K}}$, $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ and $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ is BP-complete on $\mathcal{K}$-relations for arbitrary semirings, m-semirings, and finitely generated semirings, respectively. In contrast, $\mathcal{RA}^+_{\mathcal{K}}$ was shown to be BP-complete in the standard relational case [4, 18]. We show, however, that $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$ is BP-complete on $\mathcal{K}$-relations for arbitrary finitely generated m-semirings $\mathcal{K}$.

Organization: The paper is organized as follows. After recalling in Section 2 the basic notions of $\mathcal{K}$-relations and the positive query language $\mathcal{RA}^+_{\mathcal{K}}$, we present in Section 3, the query languages $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$, $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ and $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$, obtained by extending $\mathcal{RA}^+_{\mathcal{K}}$ with *difference* and *constant annotations*. Then, in Section 4, we discuss the relationship between provenance and $\mathcal{K}$-relations, and show how the provenance semiring can be extended to record provenance for $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$, $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ and $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$. Section 5 discusses BP-completeness of $\mathcal{RA}^+_{\mathcal{K}}$ and extensions thereof. We conclude the paper in Section 6.

2. Preliminaries

In this section we recall the notions of $\mathcal{K}$-relation and the query language $\mathcal{RA}^+_{\mathcal{K}}$ that were introduced by Green et al. [12]. Then, we conclude the section by discussing an important property of $\mathcal{RA}^+_{\mathcal{K}}$, named *homomorphism property*.

$R_1 =$

drink	kind	origin	
Heineken	beer	Holland	false
Montefalco	wine	Italy	true
Pinot	grappa	Italy	true

$R_2 =$

drink	kind	origin	
Stella	beer	Belgium	2
Montefalco	wine	Italy	1
Pinot	grappa	Italy	1

$R_3 =$

drink	kind	origin	
Stella	beer	Belgium	*party*
Montefalco	wine	Italy	*tasting*
Pinot	grappa	Italy	*party*$\vee$*tasting*

$R_4 =$

drink	kind	origin	
Stella	beer	Belgium	P
Montefalco	wine	Italy	T
Pinot	grappa	Italy	$P \cup T$

Figure 1: Examples of $\mathcal{K}$-relations.

2.1. $\mathcal{K}$-relations

A (commutative) semiring $\mathcal{K} = (\mathbb{K}, \oplus, \otimes, 0, 1)$ is an algebraic structure consisting of a set $\mathbb{K}$ equipped with two binary operations, *i.e.,* sum ($\oplus$) and product ($\otimes$), such that $(\mathbb{K}, \oplus, 0)$ is a commutative monoid with identity element 0; $(\mathbb{K}, \otimes, 1)$ is a commutative monoid with identity element 1; the operation $\otimes$ distributes over $\oplus$; and finally 0 is an annihilating element. Recall that a monoid consists of set equipped with a binary operation that is associative and has an identity element. Furthermore, the set is closed under the binary operation, *i.e.,* the result of the operation on any two elements in the set belongs to the set as well.

Example 1. It is easily verified that the following structures are semirings: (1) The Boolean semiring $\mathcal{K}_{\mathbb{B}} = (\mathbb{B}, \vee, \wedge, \text{false}, \text{true})$ with $\mathbb{B} = \{\text{true}, \text{false}\}$; (2) The natural numbers $\mathcal{K}_{\mathbb{N}} = (\mathbb{N}, +, \times, 0, 1)$; (3) $\mathcal{K}_{c\text{-table}^+} = (\text{PosBool}(X), \vee, \wedge, \text{false}, \text{true})$, where $\text{PosBool}(X)$ is the set of all positive Boolean expressions (over a finite set of variables X) in which any two equivalent expressions are identified; and (4) the probabilistic semiring $\mathcal{K}_{\text{prob}} = (\mathcal{P}(\Omega), \cup, \cap, \emptyset, \Omega)$, where Ω is a finite set of events and $\mathcal{P}(\Omega)$ stands for the powerset of Ω. □

To formally introduce semirings into the relational data model, we next recall the definition of $\mathcal{K}$-relations (see [12] for more details). Let $\mathbb{D}$ be an (infinite) domain of data values and let U be a finite set of attributes. We define a *U-tuple* $\bar{t}$ to be mapping from $U \to \mathbb{D}$. The set of U-tuples is denoted by U-Tup. A *relation over U* is a subset of U-Tup. Let $\mathcal{K} = (\mathbb{K}, \oplus, \otimes, 0, 1)$ be a semiring. A *$\mathcal{K}$-relation* R over U is then a function $R : U\text{-Tup} \to \mathbb{K}$. The *support* of a $\mathcal{K}$-relation R, denoted by $\text{supp}(R)$, is defined as $\text{supp}(R) = \{\bar{t} \mid R(\bar{t}) \neq 0\}$; it is the standard relational database underlying R. The *active domain* of a $\mathcal{K}$-relation R, denoted by $\text{adom}(R)$, is defined as the set of data values (in $\mathbb{D}$) occurring in $\text{supp}(R)$.

As already mentioned in the introduction, $\mathcal{K}$-relations have recently been used to unify a variety of data models, including the standard relational model with both set and bag semantics, incomplete databases (positive Boolean c-tables to be more precise) and probabilistic databases [12].

Example 2. Consider the set of attributes $U = \{\text{drink}, \text{kind}, \text{origin}\}$. Figure 1 shows $\mathcal{K}$-relations over U, for the four different semirings described in Example 1. The semiring value associated with each tuple is shown in the last column. (1) R_1 is a $\mathcal{K}_{\mathbb{B}}$-relation and corresponds to a standard relational table with set semantics; the interpretation of this relation is that the tuple $\bar{t}_h = (\text{Heineken}, \text{beer}, \text{Holland})$ does not belong to the support of the relation while the tuples $\bar{t}_m = (\text{Montefalco}, \text{wine}, \text{Italy})$ and $\bar{t}_p = (\text{Pinot}, \text{grappa}, \text{Italy})$ do; (2) R_2 is a $\mathcal{K}_{\mathbb{N}}$-relation and corresponds to a relational table with bag semantics; the interpretation of R_2 is that there are two tuples $\bar{t}_s = (\text{Stella}, \text{beer}, \text{Belgium})$, while there is only one tuple $\bar{t}_m$ and one tuple $\bar{t}_p$; (3) R_3 is a $\mathcal{K}_{c\text{-table}^+}$ and corresponds to a (positive) Boolean c-table [13]. A Boolean c-table is a restricted form of c-tables [15] in which variables take Boolean values and only appear in conditions (not in the attributes); Hence, the tuple $\bar{t}_s$ belongs to the support of relation if there is a party, the tuple $\bar{t}_m$ belongs to it if there is a tasting, while the tuple $\bar{t}_p$ is present if there is a party or a tasting; finally, (4) R_4 is a $\mathcal{K}_{\text{prob}}$-relation and corresponds to a probabilistic event table introduced in [10, 19]; assuming that both P and T

denote probabilistic events, then the tuple $\bar{t}_s$ belongs to the support of the relation with the probability of event P, the tuple $\bar{t}_m$ with probability of event T and the tuple $\bar{t}_p$ with probability of the event $P \cup T$. □

The real strength of $\mathcal{K}$-relations becomes apparent, however, when considering provenance information. Indeed, the flexibility of semirings allows for the definition of new provenance models at different levels of granularity. We will illustrate this in more detail in Section 4 after we describe query languages on $\mathcal{K}$-relations.

2.2. The query language $\mathcal{RA}^+_{\mathcal{K}}$

The introduction of semirings in the relational model requires the redefinition of the semantics of the standard relational algebra operators. Recall that the relational algebra consists of projection, selection, union, renaming and difference [1]. When difference is omitted, one obtains the so-called positive fragment of the relational algebra or positive algebra for short. In [12], the semantics of the positive algebra on $\mathcal{K}$-relations has been introduced. We next recall the definition of the positive relational algebra on $\mathcal{K}$-relations, denoted by $\mathcal{RA}^+_{\mathcal{K}}$. As before, $\mathcal{K} = (\mathbb{K}, \oplus, \otimes, 0, 1)$ denotes a semiring. Then $\mathcal{RA}^+_{\mathcal{K}}$ includes the following operators:

empty relation For any set of attributes U, we have $\emptyset : U\text{-Tup} \to \mathbb{K}$ such that $\emptyset(\bar{t}) = 0$ for any $\bar{t}$.

union If $R_1, R_2 : U\text{-Tup} \to \mathbb{K}$ then $R_1 \cup R_2 : U\text{-Tup} \to \mathbb{K}$ is defined by

$$(R_1 \cup R_2)(\bar{t}) = R_1(\bar{t}) \oplus R_2(\bar{t}).$$

projection If $R : U\text{-Tup} \to \mathbb{K}$ and $V \subseteq U$ then $\pi_V(R) : V\text{-Tup} \to \mathbb{K}$ is defined by

$$(\pi_V R)(\bar{t}) = \bigoplus_{\bar{t} = \bar{t}' \text{ on } V \text{ and } R(\bar{t}') \neq 0} R(\bar{t}').$$

selection If $R : U\text{-Tup} \to \mathbb{K}$ and the selection predicate $\mathbf{P}$ maps each U-tuple to either 0 or 1 depending on the (in-)equality of attribute values, then $\sigma_{\mathbf{P}}(R) : U\text{-Tup} \to \mathbb{K}$ is defined by

$$(\sigma_{\mathbf{P}}(R))(\bar{t}) = R(\bar{t}) \otimes \mathbf{P}(\bar{t}).$$

natural join If $R_i : U_i\text{-Tup} \to \mathbb{K}$, for $i = 1, 2$, then $R_1 \bowtie R_2$ is the $\mathcal{K}$-relation over $U_1 \cup U_2$ defined by

$$(R_1 \bowtie R_2)(\bar{t}) = R_1(\bar{t}) \otimes R_2(\bar{t}).$$

renaming If $R : U\text{-Tup} \to \mathbb{K}$ and $\beta : U \to U'$ is a bijection then $\rho_\beta(R)$ is the $\mathcal{K}$-relation over U' defined by

$$(\rho_\beta R)(\bar{t}) = R(\bar{t} \circ \beta^{-1}).$$

It is observed in [12] that the semantics of $\mathcal{RA}^+_{\mathcal{K}}$ coincides with standard positive relational algebras for various semirings encountered in the database literature, *i.e.,* for $\mathcal{K}_{\mathbb{B}}$ (set semantics) [1], $\mathcal{K}_{\mathbb{N}}$ (bag semantics) [16], $\mathcal{K}_{c\text{-tables}^+}$ (positive Boolean c-tables under closed world semantics) [13, 15] and $\mathcal{K}_{\text{prob}}$ (probabilistic event tables) [10, 19].

2.3. The homomorphism property of $\mathcal{RA}^+_{\mathcal{K}}$

A desirable property of query languages is that they provide the user with a conceptual interface of the underlying data, independent of how exactly that data is stored and without interpreting the exact data objects [2]. This property is also known as *genericity* of query languages and is formally stated that for any permutation π of the data domain $\mathbb{D}$ and any database R, a query Q in the query language should satisfy $Q(\pi(R)) = \pi(Q(R))$. Here, π is extended to a function on tuples in the standard way. The relational algebra with set semantics is known to be generic.

A similar property, named *homomorphism property*, is desired to hold with respect to the values in the semiring for query languages over $\mathcal{K}$-relations. Intuitively, the homomorphism property ensures that the $\mathcal{RA}^+_{\mathcal{K}}$ operations do not interpret the values of the underlying semiring. Formally, let $\mathcal{K} = (\mathbb{K}, \oplus_{\mathbb{K}}, \otimes_{\mathbb{K}}, 0_{\mathbb{K}}, 1_{\mathbb{K}})$ and $\mathcal{K}' = (\mathbb{K}', \oplus_{\mathbb{K}'}, \otimes_{\mathbb{K}'}, 0_{\mathbb{K}'}, 1_{\mathbb{K}'})$ be two (commutative) semirings and let $h : \mathbb{K} \to \mathbb{K}'$ be a mapping. It is shown in [12] that the transformation from $\mathcal{K}$-relations to $\mathcal{K}'$-relations induced by h, which we also denote by h, satisfies the property that $Q(h(R)) = h(Q(R))$ for any $Q \in \mathcal{RA}^+_{\mathcal{K}}$ iff h is a semiring homomorphism [12]. That is, h satisfies the following properties: $h(0_{\mathbb{K}}) = 0_{\mathbb{K}'}$, $h(1_{\mathbb{K}}) = 1_{\mathbb{K}'}$, and for any $x, y \in \mathbb{K}$, $h(x \oplus_{\mathbb{K}} y) = h(x) \oplus_{\mathbb{K}'} h(y)$ and $h(x \otimes_{\mathbb{K}} y) = h(x) \otimes_{\mathbb{K}'} h(y)$.

3. The query languages $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$, $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ and $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$

In this section we provide three extensions of $\mathcal{RA}^+_{\mathcal{K}}$: First, we extend $\mathcal{RA}^+_{\mathcal{K}}$ with a *difference operator* ($\setminus$) resulting in the algebra $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ over $\mathcal{K}$-relations. Second, we extend $\mathcal{RA}^+_{\mathcal{K}}$ with (a family of) operators called *constant annotations* (δ). These can be thought of as a generalization of the duplicate elimination operator, an operator that is normally included in query languages over bags. The resulting query language is denoted by $\mathcal{RA}^+_{\mathcal{K}}(\delta)$. Finally, we extend $\mathcal{RA}^+_{\mathcal{K}}$ with both the difference and constant annotations, resulting in $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$.

3.1. The query language $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$

We first extend $\mathcal{RA}^+_{\mathcal{K}}$ with a difference operator. More specifically, we identify a large class of semirings that can be equipped with a so-called *monus* operator $\ominus$. The addition of the monus operator on semirings will then allow to extend $\mathcal{RA}^+_{\mathcal{K}}$ with a difference operator ($\setminus$). Finally, we show that $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ satisfies a homomorphism property similar to $\mathcal{RA}^+_{\mathcal{K}}$.

3.1.1. Semirings with monus

We follow the standard approach for introducing a "monus" operator, denoted by $\ominus$, into additive commutative monoids [3]. As we will see shortly, when introducing $\ominus$ one has to pose some restriction on the class of semirings. More specifically, we first assume that $\mathcal{K}$ is *naturally ordered*. That is, the quasi-order $x \preceq y$ on $\mathbb{K}$ defined as $x \preceq y$ iff there exists a $z \in \mathbb{K}$ such that $x \oplus z = y$, must define a *partial order* on $\mathbb{K}$. This means that apart from being reflexive and transitive, $\preceq$ should also be antisymmetric.

It is easily verified that all examples of semirings described in this paper are naturally ordered. We additionally require the following property (†): for each pair of elements $x, y \in \mathbb{K}$, the set $\{z \in \mathbb{K} \mid x \preceq y \oplus z\}$ has a smallest element. Note that the fact that $\preceq$ defines a partial order guarantees that $\{z \in \mathbb{K} \mid x \preceq y \oplus z\}$ has a *unique* smallest element, provided that it exists.

Definition 1. Let $\mathcal{K}$ be a naturally ordered semiring that satisfies property (†). For any $x, y \in \mathbb{K}$, we define $x \ominus y$ to be the smallest element z such that $x \preceq y \oplus z$. A semiring $\mathcal{K}$ which can be equipped with a monus operator $\ominus$ is called a *semiring with monus* or *m-semiring* for short.

A classical result in theory of additive commutative monoids with monus, or CMM for short, identifies two "natural" classes of CMMs [3]. Indeed, Amer shows that there are only two equationally complete classes of CMMs in the variety of CMMs. These are respectively Boolean algebras (or prime ideals thereof), for which the monus behaves like set difference, and so-called positive cones of lattice-ordered commutative groups, for which the monus behaves like the truncated minus of the natural numbers. Translated to the setting of m-semirings, this dichotomy translates to m-semirings that are a Boolean algebra on the one hand, and m-semirings that are the positive cone of a lattice-ordered commutative ring on the other hand [14, 17]. In the following example, we revisit the semirings described in Example 1 and discuss their extension to m-semirings.

Example 3. One can easily verify that the semirings described in Example 1 in Section 2 all satisfy property (†). Hence, they can *all* be extended to m-semirings. Moreover, it is easily verified that they all fall in one of the two natural classes of m-semirings described above, except for $\mathcal{K}_{c\text{-table}^+}$. More specifically, $\mathcal{K}_{\mathbb{B}}$ and $\mathcal{K}_{\text{prob}}$ are both Boolean algebras and the monus behaves like set difference. On the other hand, $\mathcal{K}_{\mathbb{N}}$ is the positive cone of the ring $\mathbb{Z}$, *i.e.,* $\mathbb{N} = \{n \mid n \in \mathbb{Z}, 0 \preceq n\}$. Consequently, the monus on $\mathcal{K}_{\mathbb{N}}$ corresponds to the truncated minus, *i.e.,* $m \ominus n = m \dot{-} n$ which is defined as $m - n$ if $m > n$ and 0 otherwise. Finally, the case of $\mathcal{K}_{\text{c-table}^+}$ is more subtle since the corresponding m-semiring is neither a Boolean algebra nor the positive cone of a lattice-ordered ring. In fact, the semiring $\mathcal{K}_{c\text{-table}^+} = (\text{PosBool}(X), \vee, \wedge, \text{false}, \text{true})$ was originally defined for positive queries only and therefore only positive Boolean expressions over X were allowed [12]. The original definition of c-tables, however, does allow for *arbitrary* Boolean expressions [15]. Hence, we define the semiring $\mathcal{K}_{c\text{-table}}$ as $(\text{Bool}(X), \vee, \wedge, \text{false}, \text{true})$, where $\text{Bool}(X)$ is the set of Boolean expressions over X in which any two equivalent expressions are identified. Clearly, $\mathcal{K}_{c\text{-table}}$ is a Boolean algebra. Furthermore, for any two expressions ϕ_1, ϕ_2 in $\text{Bool}(X)$, we have that $\phi_1 \ominus \phi_2$ is a Boolean expression that is equivalent to $\phi_1 \wedge \neg\phi_2$, as expected. □

It is not surprising that not every semiring can be extended to an m-semiring.

Example 4. From the definition of m-semiring it follows that a semiring cannot be extended to an m-semiring if the semiring is not naturally ordered or it is naturally ordered but property (†) fails to hold. For instance, consider the semiring $\mathcal{K}_{\mathbb{R}} = (\mathbb{R}, +, \times, 0, 1)$. Clearly, $r \preceq s$ for any two elements $r, s \in \mathbb{R}$ and hence $\preceq$ is not antisymmetric. Worse still, for any $r, s \in \mathbb{R}$, the set $\{t \in \mathbb{R} \mid r \preceq s + t\}$ is equal to $\mathbb{R}$ which contains no minimum since $\preceq$ holds for any two elements in $\mathbb{R}$. Therefore, $r \ominus s$ is ill-defined in $\mathcal{K}_{\mathbb{R}}$. We observe that $\mathcal{K}_{\mathbb{R}}$ is in fact a *ring* and any element $r \in \mathbb{R}$ has a unique additive inverse $-r \in \mathbb{R}$. It is possible to develop a theory of $\mathcal{K}$-relations, where $\mathcal{K}$ are ring structures, but we leave this to future work. Consider next the semiring $\mathcal{K}_{\mathbb{R}_{\min}} = (\mathbb{R} \cup \{+\infty\}, \min, +, +\infty, 0)$ where $\min\{x, y\}$ returns the minimum of x and y according to the usual ordering on $\mathbb{R} \cup \{+\infty\}$. It is easily verified $\mathcal{K}_{\mathbb{R}_{\min}}$ is naturally ordered. Indeed, if there exists a z such that $\min\{x, z\} = y$ and if in addition there exists a z' such that $\min\{y, z'\} = x$, then it follows that $x = y$. However, for any $x, y \in \mathbb{R} \cup \{+\infty\}$, the set $\{z \in \mathbb{R} \cup \{+\infty\} \mid x \preceq \min\{y, z\}\}$ is equal to $\{z \in \mathbb{R} \cup \{+\infty\} \mid \exists z' \min\{x, z'\} = \min\{y, z\}\}$. Clearly, this is not bounded below since one can take arbitrary small values for z. Hence, although $\mathcal{K}_{\mathbb{R}_{\min}}$ is naturally ordered, it does not satisfy property (†) and the monus operator is ill-defined in this semiring. We observe that a slight modification of $\mathcal{K}_{\mathbb{R}_{\min}}$ remediates the above problem. Indeed, consider $(\mathbb{R} \cup \{-\infty, +\infty\}, \min, +, +\infty, 0)$. The addition of the lowest element $-\infty$ ensures that this semiring satisfies property (†). The resulting semiring is known as the dual of the schedule algebra.

3.1.2. The difference operator

We are now ready to extend $\mathcal{RA}^+_{\mathcal{K}}$ with the difference operator. Let $\mathcal{K}$ be an arbitrary m-semiring. Then, we obtain $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ by extending $\mathcal{RA}^+_{\mathcal{K}}$ with the operator

difference If $R_1, R_2 : U\text{-Tup} \to \mathbb{K}$ then $R_1 \ominus R_2 : U\text{-Tup} \to \mathbb{K}$ is defined by

$$(R_1 \setminus R_2)(t) = R_1(t) \ominus R_2(t).$$

As a sanity check, from Example 3, it immediately follows that $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ coincides with the (full) relational algebra on relational databases for $\mathcal{K}_{\mathbb{B}}$ (set semantics), and the bag algebra with the monus operator for $\mathcal{K}_{\mathbb{N}}$ [16]. Furthermore, in the case of $\mathcal{K}_{c\text{-table}}$ it coincides with the semantics of the relational algebra on Boolean c-tables under closed world semantics [15] and for $\mathcal{K}_{\text{prob}}$ it coincides with the semantics of the relational algebra provided on probabilistic event tables [10, 19].

3.1.3. The homomorphism property for $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$

When looking at m-semirings the notion of semiring homomorphism needs to be revisited. Specifically, let $\mathcal{K} = (\mathbb{K}, \oplus_{\mathbb{K}}, \otimes_{\mathbb{K}}, \ominus_{\mathbb{K}}, 0_{\mathbb{K}}, 1_{\mathbb{K}})$ and $\mathcal{K}' = (\mathbb{K}', \oplus_{\mathbb{K}'}, \otimes_{\mathbb{K}'}, \ominus_{\mathbb{K}'}, 0_{\mathbb{K}'}, 1_{\mathbb{K}'})$ be two m-semirings. A mapping $h : \mathbb{K} \to \mathbb{K}'$ is an *m-semiring homomorphism* iff it is a semiring homomorphism and furthermore it also preserves $\ominus$, *i.e.,* for any two elements $x, y \in \mathbb{K}$ we have that $h(x \ominus_{\mathbb{K}} y) = h(x) \ominus_{\mathbb{K}'} h(y)$. The following is easily verified:

Proposition 1. *Let $\mathcal{K}$ and $\mathcal{K}'$ be two m-semirings. Let $h : \mathbb{K} \to \mathbb{K}'$ be a mapping. Then the transformation induced by h from $\mathcal{K}$-relations to $\mathcal{K}'$-relations commutes with any $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$-query Q,i.e., $Q(h(R)) = h(Q(R))$ if and only if h is an m-homomorphism.*

Proof. The proof is by induction on the structure of queries in $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$. Since $\mathcal{RA}^+_{\mathcal{K}}$ is embedded in $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ we already know by the result for $\mathcal{RA}^+_{\mathcal{K}}$ that h being a semiring homomorphism is a necessary requirement. Suppose that h is an m-homomorphism as well and that $Q = Q_1 \setminus Q_2$ (the other cases are treated as in the $\mathcal{RA}^+_{\mathcal{K}}$ case [12]). By the induction hypothesis we have that $Q(h(R)) = Q_1(h(R)) \setminus Q_2(h(R)) = h(Q_1(R)) \setminus h(Q_2(R))$. Furthermore, since h is an m-homomorphism and by the definition of $\setminus$ we have that $h(Q_1(R)(\bar{t})) \ominus_{\mathbb{K}'} h(Q_2(R)(\bar{t})) = h(Q_1(R)(\bar{t}) \ominus_{\mathbb{K}} Q_2(R)(\bar{t}))$ for all $\bar{t}$. Hence, $Q(h(R)) = h(Q(R))$.

Conversely, it is easily verified that the requirement that $Q(h(R)) = h(Q(R))$ for any R implies that h must preserve $\ominus$. That h must be a semiring homomorphism is already established in [12]. Indeed, consider $Q = (\pi_A(\sigma_{A=B}(R)) \setminus \pi_A(\sigma_{A \neq B}(R))$ and $R = \{(a,a) \mapsto x, (a,b) \mapsto y\}$ for $a \neq b$ and arbitrary $x, y \in \mathbb{K}$. Since the query result contains one tuple (a) associated with $h(x) \ominus_{\mathbb{K}'} h(y)$ in the one case and $h(x \ominus_{\mathbb{K}} y)$ in the other, the result immediately follows. □

3.2. The query language $\mathcal{RA}^+_{\mathcal{K}}(\delta)$

We next extend the positive algebra $\mathcal{RA}^+_{\mathcal{K}}$ on $\mathcal{K}$-relations with a family of operators called *constant annotations*. These operators are a generalization of the duplicate elimination operator present in most algebras over bags [16]. The intuition behind these operators is that they are "forgetful", *i.e.,* they allow to replace all values of tuples in $\mathcal{K}$-relations by some constant value. Similar to $\mathcal{RA}^+_{\mathcal{K}}$ and $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$, we show that $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ satisfies a homomorphism property.

3.2.1. Constant annotations

When considering $\mathcal{K}_{\mathbb{N}}$-relations it is common to include the duplicate elimination operator δ in the query language. Intuitively, when δ is applied on a bag-relation the result is a relation with the same support but in which each tuple is counted only once. In the language of $\mathcal{K}$-relations, $\delta(R)(\bar{t}) = 1$ for all $\bar{t}$ in $\text{supp}(R)$ and $\delta(R)(\bar{t}) = 0$ otherwise.

To introduce duplicate elimination in $\mathcal{RA}^+_{\mathcal{K}}$ on general $\mathcal{K}$-relations, we restrict our attention to m-semirings $\mathcal{K} = (\mathbb{K}, \oplus, \otimes, \ominus, 0, 1)$ that are *finitely generated*, *i.e.,* every element in $\mathbb{K}$ can be written as a

finite sequence of sums, monus and products of a finite set of elements $\mathbf{k}_1, \ldots, \mathbf{k}_m$ in $\mathbb{K}$, called *generators* of $\mathbb{K}$. We denote a set of generators of $\mathbb{K}$ by $\mathrm{Gen}(\mathbb{K})$ and for convenience assume it is minimal.

Example 5. The semirings considered so far are all finitely generated. Indeed, it is easily verified that $\mathrm{Gen}(\mathbb{B}) = \{\mathrm{true}\}$, $\mathrm{Gen}(\mathbb{N}) = \{1\}$, $\mathrm{Gen}(\mathrm{Bool}(X)) = X$, and $\mathrm{Gen}(\mathcal{P}(\Omega)) = \Omega$. The two semirings $\mathcal{K}_{\mathbb{R}}$ and $\mathcal{K}_{\mathbb{R}_{\min}}$ given in Example 4 are not finitely generated since they consist of uncountably many elements. □

We now formally define the notion of constant annotations. Given a finitely generated m-semiring $\mathcal{K} = (\mathbb{K}, \oplus, \otimes, \ominus, 0, 1)$ with generators $\mathrm{Gen}(\mathbb{K}) = \{\mathbf{k}_1, \ldots, \mathbf{k}_m\}$, we define the following set of *constant annotation* operators:

constant annotation If $R : U\text{-Tup} \to \mathbb{K}$ and $\mathbf{k}_i$ is a generator of $\mathbb{K}$ then $\delta_{\mathbf{k}_i} : U\text{-Tup} \to \mathbb{K}$ is defined by

$$(\delta_{\mathbf{k}_i}(R))(\bar{t}) = \mathbf{k_i} \text{ for each } \bar{t} \in \mathrm{supp}(R) \text{ and } (\delta_{\mathbf{k}_i}(R))(\bar{t}) = 0 \text{ otherwise.}$$

We denote by $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ the query language obtained by extending $\mathcal{RA}^+_{\mathcal{K}}$ with the constant annotation operators for the m-semiring $\mathcal{K}$ and set of generators of $\mathbb{K}$ under consideration. Note that for some m-semirings, constant annotations do not add expressive power. For instance, for $\mathcal{K}_{\mathbb{B}}$ we have that $\mathcal{RA}^+_{\mathcal{K}}(\delta) = \mathcal{RA}^+_{\mathcal{K}}$.

3.2.2. The homomorphism property for $\mathcal{RA}^+_{\mathcal{K}}(\delta)$

When considering the homomorphism property of queries in $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ one has to make the choice of generators in $\mathcal{K}$ and $\mathcal{K}'$ explicit. Let $\mathrm{Gen}(\mathbb{K}) = \{\mathbf{k}_1, \ldots, \mathbf{k}_n\}$ and $\mathrm{Gen}(\mathbb{K}') = \{\mathbf{l}_1, \ldots, \mathbf{l}_m\}$. We say that h is an *generator preserving* semiring homomorphism from $\mathcal{K}$ to $\mathcal{K}'$ if h is a semiring homomorphism and furthermore, $h(\mathrm{Gen}(\mathbb{K})) = \mathrm{Gen}(\mathbb{K}')$. Given a query $Q \in \mathcal{RA}^+_{\mathcal{K}}(\delta)$, let $h(Q)$ be the query in $\mathcal{RA}^+_{\mathcal{K}'}(\delta)$ obtained by replacing each occurrence of $\delta_{\mathbf{k}_i}$ by $\delta_{h(\mathbf{k}_i)}$. Observe that for generator preserving homomorphisms h, each $\delta_{h(\mathbf{k}_i)}$ is of the form $\delta_{\mathbf{l}_j}$ for some $j = 1, \ldots, m$. In other words, $h(Q)$ is well-defined. The following is now easily verified:

Proposition 2. *Let $\mathcal{K}$ and $\mathcal{K}'$ be two semirings with generators Gen($\mathbb{K}$) and Gen($\mathbb{K}'$), respectively. Let h be a generator-preserving homomorphism from $\mathcal{K}$ to $\mathcal{K}'$. Then, for any query $Q \in \mathcal{RA}^+_{\mathcal{K}}(\delta)$, $h(Q)(h(R)) = h(Q(R))$ where $h(Q) \in \mathcal{RA}^+_{\mathcal{K}'}(\delta)$ is the query defined above.*

3.3. The query language $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$

Finally, we introduce the query language obtained by extending $\mathcal{RA}^+_{\mathcal{K}}$ with both the difference and constant annotations operators. The resulting language is denoted by $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$ and this language also satisfies a homomorphism property. Indeed, it is easily verified that Proposition 2 carries over to this setting by replacing homomorphism by m-homomorphism and substituting the appropriate query languages.

4. $\mathcal{K}$-relations and provenance

Besides providing a general framework capturing many data models encountered in the literature, $\mathcal{K}$-relations are particularly useful for tracking various kinds of provenance information [6, 12]. We illustrate this with two examples: the *lineage semiring* and the *provenance semiring*. We refer again to Green et al. [12] and [11] for more details concerning these and other provenance models. In particular, in this section we recall how to compute the why- and how-provenance for positive queries and present m-semirings that allow for computing provenance information in the presence of difference in the relational algebra queries. We conclude this section by describing how to compute provenance in the presence of constant annotations.

$R_5 =$

drink	kind	origin	
Stella	beer	Belgium	$\{x\}$
Montefalco	wine	Italy	$\{y\}$
Pinot	grappa	Italy	$\{z\}$

$R_6 =$

drink	kind	origin	
Pinot	wine	France	$\{v\}$
Ardbeg	whiskey	Scotland	$\{w\}$

$R_7 =$

drink	kind	
Stella	beer	$\{x\}$
Montefalco	wine	$\{y\}$
Montefalco	grappa	$\{y,z\}$
Pinot	wine	$\{y,z,v\}$
Pinot	grappa	$\{z\}$
Ardbeg	whiskey	$\{w\}$

Figure 2: The lineage semiring.

$\bar{R}_5 =$

drink	kind	origin	
Stella	beer	Belgium	x
Montefalco	wine	Italy	y
Pinot	grappa	Italy	z

$\bar{R}_6 =$

drink	kind	origin	
Pinot	wine	France	v
Ardbeg	whiskey	Scotland	w

$R_8 =$

drink	kind	
Stella	beer	x^2
Montefalco	wine	y^2
Montefalco	grappa	yz
Pinot	wine	$yz+v$
Pinot	grappa	z^2
Ardbeg	whiskey	w

Figure 3: The provenance semiring.

4.1. The lineage semiring

Lineage/why-provenance was defined in [5, 9] as a way of relating the tuples in a query output to the tuples in the source relations that contribute to them. Let X be a finite set representing the ids of the tuples in the source relations. Then, the *lineage semiring* $\mathcal{K}_{\text{lin}} = (\mathcal{P}(X), \cup, \cup, \emptyset, \emptyset)$ can be used to represent and compute the why-provenance, as we illustrate in the following example.

Example 6. Consider the $\mathcal{K}_{\text{lin}}$-relations R_5, R_6 shown in Figure 2, where the set of source tuples ids is $X = \{x, y, z, v, w\}$. In both R_5 and R_6 tuples are annotated with the singleton containing their respective id. Next, let $Q(R', R'')$ be the following query over the relations R' and R'' of schema $U = \{\mathsf{drink}, \mathsf{kind}, \mathsf{origin}\}$:

$$Q(R', R'') = \pi_{\mathsf{drink},\mathsf{kind}}(\pi_{\mathsf{drink},\mathsf{origin}} R' \bowtie \pi_{\mathsf{kind},\mathsf{origin}} R') \cup \pi_{\mathsf{drink},\mathsf{kind}} R''.$$

It is easily verified that R_7 (see Figure 2) is the query result $Q(R_5, R_6)$. The $\mathcal{K}_{\text{lin}}$-values associated with the tuples in R_7 now provide their why-provenance. For example, they state that the tuple $\bar{s}_p = (\text{Pinot}, \text{wine})$ was obtained from the contribution of the tuples in R_5 and R_6 identified by y, z and v. Note, however, that why-provenance does not provide any information on the how-provenance, *e.g.,* on the way the tuple $\bar{s}_p$ was obtained. In particular, it is not possible to infer from the why-provenance information that $\bar{s}_p$ can be obtained either from joining y and z together or from v alone. □

4.2. The provenance semiring

In order to overcome the limitations of why-provenance a more powerful provenance semiring was proposed in [12]. This semiring allows to represent and compute the how-provenance of tuples in the query result. More precisely, the *(positive algebra) provenance semiring* is defined as $\mathcal{K}_{\text{prov}} = (\mathbb{N}[X], +, \times, 0, 1)$, where X is a set of source tuple ids and $\mathbb{N}[X]$ consists of all polynomials with variables taken from X and with coefficients in $\mathbb{N}$. Hence, $\mathcal{K}_{\text{prov}}$-relations consist of tuples that are annotated with polynomials. These polynomials are to be interpreted as symbolic expressions over the source tuples ids that describe how the tuples were obtained from the source. This is illustrated in the following example:

Example 7. Consider the $\mathcal{K}_{\text{prov}}$-relations $\bar{R}_5, \bar{R}_6$ and R_8 shown in Figure 3. It can be easily checked that R_8 is the query result $Q(\bar{R}_5, \bar{R}_6)$ for the query Q given in Example 6. Consider again the tuple

$R_9 =$

drink	kind	origin	
Pinot	wine	France	2
Ardbeg	whiskey	Scotland	1

$R_{10} =$

drink	kind	
Stella	beer	4
Montefalco	wine	1
Montefalco	grappa	1
Pinot	wine	3
Pinot	grappa	1
Ardbeg	whiskey	1

Figure 4: The factorization property for $\mathcal{RA}^+_{\mathcal{K}}$.

$\bar{s}_p = (\text{Pinot}, \text{wine})$. The $\mathcal{K}_{\text{prov}}$-value of $\bar{s}_p$ is the polynomial $R_8(\bar{s}_p) = yz + v$ and states that $\bar{s}_p$ can be obtained either by joining together the tuples in $\bar{R}_5$ and $\bar{R}_6$ identified by y and z or by simply using the tuple in $\bar{R}_6$ identified by v. On the contrary, the tuple $\bar{s}_m = (\text{Montefalco}, \text{grappa})$ can only be obtained by joining together the tuples identified by y and z. Clearly, the provenance information obtained by using $\mathcal{K}_{\text{prov}}$ instead of $\mathcal{K}_{\text{lin}}$ provides more information. □

A nice property of the provenance semiring is that for any semiring $\mathcal{K}$, to evaluate queries in $\mathcal{RA}^+_{\mathcal{K}}$ on $\mathcal{K}$-relations it is sufficient to know how to evaluate these queries over $\mathcal{K}_{\text{prov}}$-relations [12]. This property, called the *factorization property* for $\mathcal{RA}^+_{\mathcal{K}}$, crucially relies on the existence of a universal object in the class of semirings which in this case is precisely the provenance semiring $\mathcal{K}_{\text{prov}} = (\mathbb{N}[X], +, \times, 0, 1)$. More formally, let $\mathcal{K}$ be a semiring, R a $\mathcal{K}$-relation and $Q \in \mathcal{RA}^+_{\mathcal{K}}$. Suppose that $\text{supp}(R) = \{\bar{t}_1, \ldots, \bar{t}_k\}$ and let $X = \{x_1, \ldots, x_k\}$ be a set of tuple ids for the tuples in $\text{supp}(R)$. That is, x_i is the tuple id for tuple $\bar{t}_i$ for $i = 1, \ldots, k$. Let $\bar{R}$ be the *abstractly tagged version* of R, obtained by letting $\bar{R}(\bar{t}_i) = x_i$ for $\bar{t}_i \in \text{supp}(R)$ and $\bar{R}(\bar{t}) = 0$ otherwise. Let $\nu : X \to \mathbb{K}$ be the *valuation* that maps x_i to $R(\bar{t}_i)$.

Because $\mathcal{K}_{\text{prov}} = (\mathbb{N}[X], +, \times, 0, 1)$ is the free semiring generated by X, we have the property that there exists a unique semiring homomorphism $\mathsf{Eval}_\nu : \mathbb{N}[X] \to \mathbb{K}$ such that for one-variable monomials we have that $\mathsf{Eval}_\nu(x) = \nu(x)$. Combined with the homomorphism property for $\mathcal{RA}^+_{\mathcal{K}}$ (see Section 2.3) and observing that $\mathsf{Eval}_\nu(\bar{R}) = R$ we recall from [12] that

$$Q(R) = \mathsf{Eval}_\nu \circ Q(\bar{R}).$$

In other words, the semantics of queries in $\mathcal{RA}^+_{\mathcal{K}}$ over arbitrary semirings *factors through* its semantics in the provenance semiring.

Example 8. Consider the $\mathcal{K}_{\text{lin}}$-relations R_5 and R_6 shown in Figure 2. Their respective abstractly tagged versions $\bar{R}_5$ and $\bar{R}_6$ are shown in Figure 3. Consider again the query Q of Example 6. Then, the $\mathcal{K}_{\text{prov}}$-relation R_8 is the query result $Q(\bar{R}_5, \bar{R}_6)$. Let ν be the valuation that maps η to $\{\eta\}$, for $\eta \in \{x, y, z, v, w\}$. The factorization property then tells us that the $\mathcal{K}_{\text{lin}}$-relation R_7, shown in Figure 2, is equal to $\mathsf{Eval}_\nu(R_8)$. Indeed, consider the tuple $\bar{s}_p = (\text{Pinot}, \text{grappa})$ annotated with $yz + v$. Then, $\mathsf{Eval}_\nu(yz + v) = (\nu(y) \cup \nu(z)) \cup \nu(v) = \{y, z, v\}$, as desired. Similarly, consider the $\mathcal{K}_{\mathbb{N}}$-relations R_2 shown in Figure 1 and R_9 shown in Figure 4. Their abstractly tagged versions $\bar{R}_2$ and $\bar{R}_9$ are identical to $\bar{R}_5$ and $\bar{R}_6$, respectively. Let ν be the valuation that maps x and v to 2 and y, z and w to 1. Then the factorization property tells that $Q(R_2, R_9) = R_{10}$, shown in Figure 4, is equal to $\mathsf{Eval}_\nu(R_8)$. Indeed, consider again the tuple $\bar{s}_p$ associated with $yz + v$. In this case we have that $\mathsf{Eval}_\nu(yz + v) = (\nu(y) \times \nu(z)) + \nu(v) = 1 + 2 = 3$, as desired. □

4.3. The provenance semiring with monus

We next describe how to represent and compute why and how provenance in the presence of difference. It is easily verified that both $\mathcal{K}_{\text{lin}}$ and $\mathcal{K}_{\text{prov}}$ can be extended to m-semirings:

$R_{10} =$

drink	kind	origin	
Stella	beer	Belgium	2
Montefalco	wine	Italy	0
Pinot	grappa	Italy	0

$R_{11} =$

drink	kind	origin	
Stella	beer	Belgium	x^2
Montefalco	wine	Italy	y^2
Pinot	grappa	Italy	z^2

$R_{12} =$

drink	kind	origin	
Stella	beer	Belgium	4
Montefalco	wine	Italy	1
Pinot	grappa	Italy	1

Figure 5: The failure of the factorization property for $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ and $\mathcal{K}_{\text{prov}'}$.

Example 9. In the case of $\mathcal{K}_{\text{lin}}$ the monus operator simply coincides with set difference. For the provenance semiring, let $X = \{x_1, \ldots, x_n\}$ be the set of variables and for $\alpha \in \mathbb{N}^n$, denote by x^α the monomial $x_1^{\alpha_1} x_2^{\alpha_2} \cdots x_n^{\alpha_n}$, where by definition $x_i^0 = 1$. Let I be a finite subset of $\mathbb{N}^n$ and let $f[X] = \sum_{\alpha \in I} f_\alpha x^\alpha$ and $g[X] = \sum_{\alpha \in I} g_\alpha x^\alpha$ be two polynomials in $\mathbb{N}[X]$. Then it is easily verified that $f[X] \ominus g[X] = \sum_{\alpha \in I} (f_\alpha \dot{-} g_\alpha) x^\alpha$, where $\dot{-}$ denotes the truncated minus on $\mathbb{N}$. □

Unfortunately, the m-semiring $\mathcal{K}_{\text{prov}'} = (\mathbb{N}[X], +, \times, \ominus, 0, 1)$ is not the universal object in the variety of all m-semirings and as a consequence it does not satisfy the factorization property for $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$:

Example 10. Let R_2 be the $\mathcal{K}_{\mathbb{N}}$-relation shown in Figure 1 and consider the query

$$Q'(R) = (R \bowtie R) - R.$$

It is easily verified that $Q'(R_2)$ is the $\mathcal{K}_{\mathbb{N}}$-relation R_{10} shown in Figure 5. The straightforward generalization of the factorization property to $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ and using $\mathcal{K}_{\text{prov}'}$ as factoring m-semiring would imply that $Q'(R_2)$ can be obtained from the query evaluation $Q'(\bar{R}_2)$ on the abstractly tagged version of R_2 (now interpreted as a $\mathcal{K}_{\text{prov}'}$-relation) and from the valuation ν that maps x to 2, and y, z to 1. The $\mathcal{K}_{\text{prov}'}$-relation $Q'(\bar{R}_2)$ is shown as relation R_{11} in Figure 5. Here, each tuple is associated with $\eta^2 \ominus \eta = (0 \cdot \eta + 1 \cdot \eta^2) \ominus (1 \cdot \eta + 0 \cdot \eta^2) = (0 \dot{-} 1) \cdot \eta + (1 \dot{-} 0) \cdot \eta^2 = \eta^2$, for some id $\eta \in \{x, y, z\}$. Then, $Q'(R_2) = R_{10} \neq \mathsf{Eval}_\nu(R_{11}) = R_{12}$. It is easily verified that a similar counterexample works when we consider the $\mathcal{K}_{\mathbb{B}}$-relation R_1 shown in Figure 1 and query Q'. Indeed, in this case $Q'(R_1)$ returns the empty relation, *i.e.,* all tuples are associated with false. On the contrary, if we consider the valuation ν maps x to false, and y, z to true, then we have that $\mathsf{Eval}_\nu(Q'(\bar{R}_1))$ contains two tuples associated with $\nu(y^2) = \nu(y) \wedge \nu(y) = \text{true}$ and $\nu(z^2) = \nu(z) \wedge \nu(z) = \text{true}$, respectively. □

We next show how a factorization property for $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ can be obtained. Indeed, from universal algebra it follows that there exists a unique free m-semiring. We next describe the construction of this semiring and then show how it can be used to represent and compute provenance for $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$.

First, we observe that the class of m-semirings is an equational variety. Indeed, an algebraic structure $(\mathbb{K}, \oplus, \otimes, \ominus, 0, 1)$ is an m-semiring iff it satisfies (i) the defining equations of being a semiring; and (ii) the defining equations of $(\mathbb{K}, \oplus, \ominus, 0)$ being a commutative monoid with monus [3]. Hence, by Birkhoff's Theorem the class of m-semirings is indeed a variety and furthermore admits free objects [7].

We recall the standard universal algebra construction for the unique free object $T[X]$ generated by $X = \{x_1, \ldots, x_n\}$ in the equational variety of m-semirings [7]. In a nutshell, elements of $T[X]$ consist of terms constructed inductively as follows: x_i, 1 and 0 are terms; and moreover, if t and s are terms then so are $(t \oplus s)$, $(t \ominus s)$ and $(t \otimes s)$; and finally, nothing else is a term.

We next need the notion of congruence relation. A congruence relation C over $T[X]$ is an equivalence relation over $T[X]$ that is compatible with $\oplus$, $\otimes$ and $\ominus$, *i.e.,* if $C(s_1, t_1)$ and $C(s_2, t_2)$ then also

$C(s_1 \,\mathrm{op}\, s_2, t_1 \,\mathrm{op}\, t_2)$ for $\mathrm{op} \in \{\oplus, \otimes, \ominus\}$. We next specialize C to correspond to the congruence relation that identifies terms based on the equations of m-semirings. It is then easily verified that the quotient structure $T[X]/C$ that consists of expressions in $T[X]$ in which any two equivalent expressions are identified (as specified by C), is indeed an m-semiring. Furthermore, it follows that $T[X]/C$ is the free m-semiring generated by X [7]. Hence, for any m-semiring $\mathcal{K}$ and any valuation $\nu : X \to \mathbb{K}$, we have that ν can be lifted to an m-semiring homomorphism $\mathsf{Eval}_\nu : T[X]/C \to \mathbb{K}$ that coincides with ν on X. We denote by $\mathcal{K}_{\mathrm{dprov}}$ the free m-semiring $(T[X]/C, \oplus, \otimes, \ominus, 0, 1)$ obtained in this way.

The following example illustrates $\mathcal{K}_{\mathrm{dprov}}$ and its corresponding factorization property.

Example 11. Consider again the relation $\bar{R}_2$ (which is equal to $\bar{R}_5$ shown in Figure 3). This can obviously be seen as a $\mathcal{K}_{\mathrm{dprov}}$ relation. Let Q' be the query of Example 10. It is easily verified that the $\mathcal{K}_{\mathrm{dprov}}$-relation $Q'(\bar{R}_2)$ is similar to the relation R_{11} shown in Figure 5, except that each tuple is now associated with $(\eta \otimes \eta) \ominus \eta$ for $\eta \in \{x, y, z\}$. If we consider the valuation ν that maps x to 2 and y, z to 1 and extend ν to an m-homomorphism $\mathsf{Eval}_\nu : T[X]/C \to \mathbb{N}$ in the natural way, then $Q'(R_2) = R_{10} = \mathsf{Eval}_\nu(Q'(\bar{R}_2))$. Indeed, this follows from the fact that $\mathsf{Eval}_\nu((\eta \otimes \eta) \ominus \eta) = (\nu(\eta) \times \nu(\eta)) \dot{-} \nu(\eta)$. Similarly, if we consider the valuation ν that maps x to false and y, z to true and let $\mathsf{Eval}_\nu : T[X]/C \to \mathbb{B}$, then $Q'(R_1) = \mathsf{Eval}_\nu(Q'(\bar{R}_1))$. This follows again from the fact that $\mathsf{Eval}_\nu((\eta \otimes \eta) \ominus \eta) = (\nu(\eta) \wedge \nu(\eta)) \ominus \nu(\eta) = \nu(\eta) \wedge \bar{\nu}(\eta) = \text{false}$, for $\eta \in \{x, y\}$. □

The following proposition is an immediate consequence of Proposition 2 and the fact that $\mathcal{K}_{\mathrm{dprov}}$ is a free m-semiring over X:

Proposition 3. *Le $\mathcal{K}$ be an m-semiring. For any query $Q \in \mathcal{RA}^+_{\mathcal{K}}(\backslash)$ and any $\mathcal{K}$-relation R with tuple id set X, $Q(R) = \mathsf{Eval}_\nu \circ Q(\bar{R})$, where $\bar{R}$ denotes the $\mathcal{K}_{dprov}$-relation obtained by tagging each tuple in R with its own tuple id.*

4.4. The provenance semiring with monus and constant annotations

We can easily extend the construction of the provenance m-semiring $\mathcal{K}_{\mathrm{dprov}}$ to obtain an extended provenance m-semiring for $\mathcal{RA}^+_{\mathcal{K}}(\backslash, \delta)$ for which a factorization property holds. We first note that the provenance semirings discussed in this and other papers [12, 11] are all finitely generated. Similarly for the extended provenance m-semiring described next.

In a nutshell, this m-semiring is constructed in the same way as $\mathcal{K}_{\mathrm{dprov}}$, with the proviso that if t is a term of the m-semiring, then so are $\delta_{\mathbf{y}_i}(t)$ for $y_i \in Y$. Here, Y is a set of variables disjoint from X. Intuitively, the factorization property holds also for $\mathcal{RA}^+_{\mathcal{K}}(\backslash, \delta)$, after extending the valuation also to variables in Y. Formally, let $\mathcal{K}$ be a finitely generated m-semiring with $\mathrm{Gen}(\mathbb{K}) = \{\mathbf{k}_1, \ldots, \mathbf{k}_n\}$. Let R be $\mathcal{K}$-relation and Q be a query in $\mathcal{RA}^+_{\mathcal{K}}(\backslash, \delta)$. Let Y be a set of n fresh variables y_i, one for each generator in $\mathcal{K}$, and let ν be the valuation of $X \cup Y$ that maps, as before, x_i to $R(\bar{t}_i)$ and y_i to $\mathbf{k}_i$. Furthermore, we define Q' to be Q in which each occurrence of $\delta_{\mathbf{k}_i}$ is replaced by δ_{y_i}. Then, $Q(R) = \mathsf{Eval}_\nu \circ Q'(\bar{R})$ where $\bar{R}$ is viewed as an extended provenance m-semiring relation.

5. BP-Completeness for $\mathcal{K}$-relations

In this section, we initiate our study of the completeness of query languages over $\mathcal{K}$-relations in the sense of Bancilhon and Paredaens [4, 18]. First, recall that Codd qualified a query language on standard relational databases as *complete* if its expressive power is at least that of the relational calculus [8]. Bancilhon [4] and Paredaens [18] independently provided a *language-independent* characterization of completeness. This characterization, now known as *BP-completeness*, can be stated as follows: a relation T is the result of

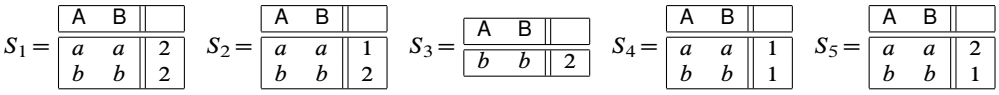

Figure 6: Example $\mathcal{K}_{\mathbb{N}}$-relations.

a *generic* relational algebra query applied to a database S if and only if (i) the active domain of T is included in the domain of S; and (ii) every automorphism of S is also an automorphism of T. In fact, Paredaens [18] observed that once inequality conditions are allowed in the selection predicate, one does not require difference in the relational algebra to be BP-complete.

Recall that a generic query is one which is oblivious to the constants appearing in the relation, *i.e.,* for any permutation τ of the domain $\mathbb{D}$, we have that $Q(\tau(R)) = \tau(Q(R))$. Furthermore, an automorphism of a relation R is a permutation τ of $\mathbb{D}$ that leaves R invariant, *i.e.,* for any $\bar{t} \in R$, $\tau(\bar{t}) \in R$. Hence, intuitively, the set of automorphisms of a relation S, denoted by $\mathrm{Aut}(S)$, allows to identify values that are "indistinguishable" for the relation, i.e. values that can be switched without changing the relation itself.

In order to study BP-completeness in the setting of $\mathcal{K}$-relations, we first need to define the notion of *automorphism of a $\mathcal{K}$-relation*. Given that $\mathcal{K}$-relations are annotated relations, by analogy to the case of standard relations, $\mathcal{K}$-relations should allow to identify values in the support that can be switched without changing neither the tuples, nor the respective tuples annotations. That is, apart from being an automorphism of the underlying relational database, an automorphism of a $\mathcal{K}$-relation should additionally preserve the semiring values associated with the tuples. Hence, formally, the set of *automorphisms* of R, denoted by $\mathrm{Aut}_{\mathcal{K}}(R)$, is defined as

$$\mathrm{Aut}_{\mathcal{K}}(R) = \{\tau \mid \tau \in \mathrm{Aut}(\mathrm{supp}(R)) \text{ and } R(\tau(\bar{t})) = R(\bar{t}), \forall \bar{t} \in \mathbb{D}^n\}.$$

Example 12. Consider the relations given in Figure 6 and assume that $\mathbb{D} = \{a, b\}$. When considering the underlying standard relations, *i.e.,* ignoring the annotations, we have that $\mathrm{Aut}(S_1) = \mathrm{Aut}(S_2) = \mathrm{Aut}(S_4) = \mathrm{Aut}(S_5) = \{(a \mapsto a, b \mapsto b), (a \mapsto b, b \mapsto a)\}$ and $\mathrm{Aut}(S_3) = \{(a \mapsto a, b \mapsto b)\}$. When viewed as $\mathcal{K}_{\mathbb{N}}$-relations, however, with the multiplicities of each tuple shown in the last column, we have that $\mathrm{Aut}_{\mathcal{K}}(S_1) = \mathrm{Aut}_{\mathcal{K}}(S_4) = \{(a \mapsto a, b \mapsto b), (a \mapsto b, b \mapsto a)\}$, $\mathrm{Aut}_{\mathcal{K}}(S_2) = \mathrm{Aut}_{\mathcal{K}}(S_5) = \{(a \mapsto a, b \mapsto b)\}$ and finally, $\mathrm{Aut}_{\mathcal{K}}(S_3) = \{(a \mapsto a, b \mapsto b)\}$. □

The set of $\mathcal{K}$-relations that are *preserved by* $\mathrm{Aut}_{\mathcal{K}}(R)$, denoted by $\mathrm{Inv}_{\mathbb{D}}(R)$, is defined as:

$$\mathrm{Inv}_{\mathbb{D}}(R) = \{S \mid \mathrm{adom}(S) \subseteq \mathrm{adom}(R), \mathrm{Aut}_{\mathcal{K}}(R) \subseteq \mathrm{Aut}_{\mathcal{K}}(S)\}.$$

Example 13. Consider again the relations given in Figure 6. From the definition above, it follows that $\mathrm{Inv}_{\mathbb{D}}(S_1) = \mathrm{Inv}_{\mathbb{D}}(S_4) \subseteq \mathrm{Inv}_{\mathbb{D}}(S_2) = \mathrm{Inv}_{\mathbb{D}}(S_5)$ and moreover, $\mathrm{Inv}_{\mathbb{D}}(S_3) \subseteq \mathrm{Inv}_{\mathbb{D}}(S_i)$ for $i \in \{2, 5\}$. In particular, $S_3 \in \mathrm{Inv}_{\mathbb{D}}(S_i)$ for $i \in \{2, 5\}$. □

Finally, the expressiveness of a query language can be described in terms of the "information" that can be deduced from a $\mathcal{K}$-relation using queries in that query language. Following Paredaens [18] we define: Let $\mathcal{Q}$ be a query language and R a $\mathcal{K}$-relation, then the *basic information* of R with respect to $\mathcal{Q}$ is the set of $\mathcal{K}$-relations:

$$\mathrm{BI}(R, \mathcal{Q}) = \{S \mid Q(R) = S \text{ for some generic query } Q \in \mathcal{Q}\}.$$

Finally, BP-completeness links the notions of basic information and invariant relations together:

Definition 2. A query language $\mathcal{Q}$ is *BP-complete* if $\mathrm{BI}(R, \mathcal{Q}) = \mathrm{Inv}_{\mathbb{D}}(R)$ for all $\mathcal{K}$-relations R.

It is worth noting that the above definitions coincide with the standard notions in the relational setting under the set semantics, *i.e.,* when considering $\mathcal{K} = \mathcal{K}_{\mathbb{B}}$.

We first study BP-completeness for $\mathcal{RA}^+_{\mathcal{K}}$. A straightforward induction on the structure of queries in $\mathcal{RA}^+_{\mathcal{K}}$ shows that the inclusion of $\mathrm{BI}(R, \mathcal{RA}^+_{\mathcal{K}}) \subseteq \mathrm{Inv}_{\mathbb{D}}(R)$ holds for any semiring $\mathcal{K}$ and $\mathcal{K}$-relation R:

Lemma 1. *For any semiring $\mathcal{K}$, any $Q \in \mathcal{RA}^+_{\mathcal{K}}$ and any $\mathcal{K}$-relation R, we have that (i) $adom(Q(R)) \subseteq adom(R)$ and (ii) $Aut_{\mathcal{K}}(R) \subseteq Aut_{\mathcal{K}}(Q(R))$.*

The other direction, *i.e.,* whether $\mathrm{Inv}_{\mathbb{D}}(R) \subseteq \mathrm{BI}(R, \mathcal{RA}^+_{\mathcal{K}})$ holds for any semiring $\mathcal{K}$ and $\mathcal{K}$-relation R is not true. Indeed, a counterexample can be found for the semiring $\mathcal{K}_{\mathbb{N}}$.

Proposition 4. *There exists a semiring $\mathcal{K}$ such that $\mathcal{RA}^+_{\mathcal{K}}$ is not BP-complete on $\mathcal{K}$-relations.*

Proof. Let $\mathcal{K}$ be the semiring $\mathcal{K}_{\mathbb{N}}$ and consider the relations S_1 and S_4 in Figure 6. From Example 13 we know that $S_4 \in \mathrm{Inv}_{\mathbb{D}}(S_1)$. However, it is easily verified, by induction on the structure of queries, that for every *generic* query $Q \in \mathcal{RA}^+_{\mathcal{K}}$, $Q(S_1)$ is either: *(i)* empty, or *(ii)* the empty tuple, or *(iii)* such that it contains only tuples having *even multiplicity*. In other words, S_4 cannot be the result of a generic query in $\mathcal{RA}^+_{\mathcal{K}}$, i.e. $S_4 \notin \mathrm{BI}(S_1, \mathcal{RA}^+_{\mathcal{K}})$. □

In fact, the counterexample in the previous proof shows that more expressive power is needed to make $\mathcal{RA}^+_{\mathcal{K}}$ BP-complete for the semiring $\mathcal{K}_{\mathbb{N}}$. It is easy to see that considering the language $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ obtained by adding constant annotation operators to $\mathcal{RA}^+_{\mathcal{K}}$, resolves the previous counterexample for $\mathcal{K}_{\mathbb{N}}$. Indeed, $S_4 = \delta_{\mathbf{1}}(S_1)$ and therefore $S_4 \in \mathrm{BI}(S_1, \mathcal{RA}^+_{\mathcal{K}}(\delta))$ for $\mathcal{K}_{\mathbb{N}}$. It turns out, however, that the query language $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ is still not BP-complete for arbitrary finitely generated semirings.

Proposition 5. *There exists a finitely generated semiring $\mathcal{K}$ such that $\mathcal{RA}^+_{\mathcal{K}}(\delta)$ is not BP-complete on $\mathcal{K}$-relations.*

Proof. Let $\mathcal{K}$ be the semiring $\mathcal{K}_{\mathbb{N}}$ and consider the relations S_2 and S_5 in Figure 6. From Example 13 we know $S_5 \in \mathrm{Inv}_{\mathbb{D}}(S_2)$. It is easily verified, however, that for any *generic* query $Q \in \mathcal{RA}^+_{\mathcal{K}}(\delta)$, the query result $Q(S_2)$ satisfies the property that for any two tuples $\bar{t}_1$ and $\bar{t}_2$ in $Q(S_2)$, $\bar{t}_1$ occurs with less or equal multiplicity than $\bar{t}_2$ if and only if $\bar{t}_1$ contains a less or equal number of b's than $\bar{t}_2$. Hence, $S_5 \notin \mathrm{BI}(S_2, \mathcal{RA}^+_{\mathcal{K}}(\delta))$. □

The counterexample in the previous proof can, however, be resolved when considering $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$ instead of $\mathcal{RA}^+_{\mathcal{K}}(\delta)$. Indeed, it is easily verified that for the m-semiring $\mathcal{K}_{\mathbb{N}} = (\mathbb{N}, +, \times, \dot{-}, 0, 1)$,

$$S_5 = (((\delta_{\mathbf{1}}(S_2) \cup \delta_{\mathbf{1}}(S_2)) \setminus S_2) \cup \delta_{\mathbf{1}}(S_2).$$

In other words, in this case, we have that $S_5 \in \mathrm{BI}(S_2, \mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta))$.

At this point, one may wonder whether the extension of $\mathcal{RA}^+_{\mathcal{K}}$ with difference alone, *i.e.,* $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ results in a BP-complete language over arbitrary m-semirings. The proof of Proposition 4, however, carries through for $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$. Hence, $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ is not BP-complete for arbitrary m-semirings.

We next show that the fact $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$ resolves both counterexamples given in the proofs of Propostion 4 and 5 is not a coincidence.

Theorem 1. *The query language $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$ is BP-complete on $\mathcal{K}$-relations for all finitely generated m-semirings $\mathcal{K}$.*

Proof. We first observe that Lemma 1 extends to $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$ for any finitely generated m-semiring $\mathcal{K}$ and any $\mathcal{K}$-relation R. Indeed, a straightforward induction on the queries in $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$ shows that $\mathrm{BI}(R,\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)) \subseteq \mathrm{Inv}_{\mathbb{D}}(R)$ for any $\mathcal{K}$-relation R.

For the opposite direction, *i.e.,* given a $\mathcal{K}$-relation R, whether $\mathrm{Inv}_{\mathbb{D}}(R) \subseteq \mathrm{BI}(R,\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta))$ holds, we show that for any $\mathcal{K}$-relation $S \in \mathrm{Inv}_{\mathbb{D}}(R)$, there exists a generic query $Q \in \mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$ such that $Q(R) = S$. In other words, we show that $S \in \mathrm{BI}(R,\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta))$.

Let R and S be $\mathcal{K}$-relations and assume that $S \in \mathrm{Inv}_{\mathbb{D}}(R)$. The desired query $Q \in \mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$ such that $Q(R) = S$ is constructed in a number of steps:

First, we define a query $Q_{\mathrm{Aut}} \in \mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$ such that $Q_{\mathrm{Aut}}(R) = \mathrm{Aut}_{\mathcal{K}}(R)$. Strictly speaking, $Q_{\mathrm{Aut}}(R)$ returns permutations of $\mathrm{adom}(R)$ instead of permutation of $\mathbb{D}$. This, however, is sufficient for our purpose since $\mathrm{adom}(S) \subseteq \mathrm{adom}(R)$. More specifically, let $(a_1,\ldots,a_n)$ be a tuple that represents all values in $\mathrm{adom}(R)$. Then $Q_{\mathrm{Aut}}(R)$ consists of all tuples $(b_1,\ldots,b_n) \in \mathrm{adom}(R)^n$ such that the mapping $\tau(a_i) = b_i$, for $i = 1,\ldots,n$, is an automorphism of R, *i.e.,* $\tau \in \mathrm{Aut}_{\mathcal{K}}(R)$. Observe that $(a_1,\ldots,a_n)$ is always present in $Q_{\mathrm{Aut}}(R)$ since it corresponds to the trivial automorphism of R. The query $Q_{\mathrm{Aut}}(R)$ is constructed as follows. Assume that $\mathrm{supp}(R) = \{\bar{t}_1,\ldots,\bar{t}_p\}$ and denote the corresponding $\mathcal{K}$-values by $\ell_i = R(\bar{t}_i)$, for $i = 1,\ldots,p$. For $i = 1,\ldots,p$, we construct the following queries:

- Q_{ℓ_i}: A query such that $Q_{\ell_i}(R)(\bar{t}) = \ell_i$ for all $\bar{t} \in \mathrm{supp}(R)$ and $Q_{\ell_i}(R)(\bar{t}) = 0$ otherwise. This query can be expressed in $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$ using the constant annotation operators; Indeed, these operators allow to generate arbitrary $\mathcal{K}$-values and assign them to tuples in a relation. In particular, one can assign each tuple in R the constant value ℓ_i.

- $Q_{=\ell_i}$: A query such that $Q_{=\ell_i}(R)(\bar{t}) = \ell_i$ if $R(\bar{t}) = \ell_i$ and $Q_{=\ell_i}(R)(\bar{t}) = 0$ otherwise. That is, this query extracts all tuples $\bar{t}$ from R that satisfy $R(\bar{t}) = \ell_i$. This query is expressible in $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$. Indeed, we claim that $Q_{=\ell_i} = (R \setminus Q_{\ell_i \not\preceq}) \setminus Q_{\not\preceq \ell_i}$, where $Q_{\ell_i \not\preceq} = R \bowtie Q_{\mathbf{1}}(Q_{\ell_i}(R) \setminus R)$ and $Q_{\not\preceq \ell_i} = R \bowtie Q_{\mathbf{1}}(R \setminus Q_{\ell_i}(R))$. Here, Q_{ℓ_i} is the query previously constructed and $Q_{\mathbf{1}}$ is a similar query except that it replaces all semiring values of the tuples with the unit value $\mathbf{1}$ in $\mathcal{K}$ instead of with ℓ_i. To show the correctness of this query we first observe that $R(\bar{t}) = \ell_i$ iff both $R(\bar{t}) \preceq \ell_i$ and $\ell_i \preceq R(\bar{t})$. This follows from the fact we consider m-semirings for which the natural order $\preceq$ is, by assumption, a partial order. Consider first the query $Q_{\ell_i \not\preceq}$. It is easily verified that $Q_{\ell_i \not\preceq}(R)(\bar{t}) = R(\bar{t})$ if $\ell_i \not\preceq R(\bar{t})$ and $Q_{\ell_i \not\preceq}(R)(\bar{t}) = 0$ otherwise. Similarly, $Q_{\not\preceq \ell_i}(R)(\bar{t}) = R(\bar{t})$ if $R(\bar{t}) \not\preceq \ell_i$ and $Q_{\not\preceq \ell_i}(R)(\bar{t}) = 0$ otherwise. As a consequence, $(R \setminus Q_{\ell_i \not\preceq})(\bar{t}) = R(\bar{t})$ if $\ell_i \preceq R(\bar{t})$ and $(R \setminus Q_{\preceq \ell_i})(\bar{t}) = 0$ otherwise. From this and the definition of $Q_{\not\preceq \ell_i}$ the correctness of $Q_{=\ell_i}$ then follows.

- Q^s_{Aut}: A query that computes the automorphisms of domain values in $\mathrm{adom}(R)$ of a $\mathcal{K}$-relation in which each tuple is assigned the same $\mathcal{K}$-value. Observe that for such $\mathcal{K}$-relations T, $\mathrm{Aut}_{\mathcal{K}}(T) = \mathrm{Aut}(\mathrm{supp}(T))$. As a consequence, the query Q^s_{Aut} can be expressed in the same way as for the classical relational case [18].

Finally, we obtain $Q_{\mathrm{Aut}}(R)$ by taking the intersection of $Q^s_{\mathrm{Aut}}(Q_{=\ell_i}(R))$ for $i = 1,\ldots,p$. That is,

$$Q_{\mathrm{Aut}}(R) = Q^s_{\mathrm{Aut}}(Q_{=\ell_1}(R)) \bowtie \cdots \bowtie Q^s_{\mathrm{Aut}}(Q_{=\ell_p}(R)).$$

It is easily verified that $Q_{\mathrm{Aut}}(R) = \mathrm{Aut}_{\mathcal{K}}(R)$, as desired.

We next proceed as follows: First we define a query Q^s in $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$ such that $\mathrm{supp}(Q^s(R)) = \mathrm{supp}(S)$, *i.e.,* $Q^s(R)$ and S agree as standard relations; Second, we show how Q^s can be modified into a query Q in $\mathcal{RA}^+_{\mathcal{K}}(\setminus,\delta)$ such that $Q(R) = S$, *i.e.,* $Q(R)$ and S also agree as $\mathcal{K}$-relations.

- By assumption we have that $\text{adom}(S) \subseteq \text{adom}(R)$. Furthermore, recall that $\text{Aut}_{\mathcal{K}}(R)$ contains a tuple $(a_1, \ldots, a_n)$ that represents all values in $\text{adom}(R)$. Let $\bar{s} \in \text{supp}(S)$. It is clear that $\bar{s} \in \text{supp}(\tilde{\pi}_{\bar{s}}(Q_{\text{Aut}}(R)))$, where $\tilde{\pi}_{\bar{s}}$ stands for an appropriate generalized projection. Recall that a generalized projection is a projection in which the same attribute can be repeated several times. This operator can be simulated using the standard projection and join operator and therefore does not add to the expressive power of $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$. For instance, suppose that $\text{adom}(R) = \{a, b, c\}$ is represented by the tuple (a, b, c) in $\text{Aut}_{\mathcal{K}}(R)$. Furthermore, assume that $\bar{s} = (a, b, b)$. Then $\bar{s} \in \text{supp}(\tilde{\pi}_{1,2,2}(Q_{\text{Aut}}(R)))$. By assumption we also have that $\text{Aut}_{\mathcal{K}}(R) \subseteq \text{Aut}_{\mathcal{K}}(S)$. As a consequence, for each $\bar{s} \in \text{supp}(S)$ we have that $\text{supp}(\tilde{\pi}_{\bar{s}}(Q_{\text{aut}}(R))) \subseteq \text{supp}(S)$. In other words, $\text{supp}(S) = \bigcup_{\bar{s} \in S} \text{supp}(\tilde{\pi}_{\bar{s}}(Q_{\text{Aut}}(R)))$. Finally, we observe that for any two tuples $\bar{s}, \bar{t} \in S$, if $\bar{s} \in \text{supp}(\tilde{\pi}_{\bar{t}}(Q_{\text{Aut}}(R)))$ then $\text{supp}(\tilde{\pi}_{\bar{t}}(Q_{\text{Aut}}(R))) = \text{supp}(\tilde{\pi}_{\bar{s}}(Q_{\text{Aut}}(R)))$. As a consequence, $\text{supp}(S)$ can be partitioned as

$$\text{supp}(S) = \text{supp}(\tilde{\pi}_{\bar{s}_1}(Q_{\text{Aut}}(R))) \uplus \cdots \uplus \text{supp}(\tilde{\pi}_{\bar{s}_r}(Q_{\text{Aut}}(R))),$$

for some tuples $\bar{s}_i \in \text{supp}(S)$, $i = 1, \ldots, r$. We define $Q^s = \bigcup_{i=1}^{r} \tilde{\pi}_{\bar{s}_i}(Q_{\text{Aut}}(R))$. Clearly this query satisfies $\text{supp}(S) = \text{supp}(Q^s(R))$.

- We next show how to modify Q^s into Q such that $S = Q(R)$. Observe that it only remains to correctly set the $\mathcal{K}$-values of the tuples in $Q^s(R)$. For this, we observe that since $\text{Aut}_{\mathcal{K}}(R) \subseteq \text{Aut}_{\mathcal{K}}(S)$, we have that for each $i = 1, \ldots, r$, all tuples in $\text{supp}(\tilde{\pi}_{\bar{s}_i}(Q_{\text{aut}}(R)))$ have the same $\mathcal{K}$-value in S, say μ_i. We therefore use the constant annotation operators in $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$ to set, for each $i = 1, \ldots, r$, the $\mathcal{K}$-value of the tuples in $\tilde{\pi}_{\bar{s}_i}(Q_{\text{aut}}(R)))$ to μ_i. It is now easily verified that the query $Q = Q_{\mu_1}(\tilde{\pi}_{\bar{s}_i}(Q_{\text{aut}}(R)))) \cup \cdots \cup Q_{\mu_r}(\tilde{\pi}_{\bar{s}_r}(Q_{\text{aut}}(R))))$ satisfies $Q(R) = S$, *i.e.,* Q is the desired query. □

It is interesting to observe that in case of $\mathcal{K}_{\mathbb{B}} = (\mathbb{B}, \vee, \wedge, \ominus, \text{false}, \text{true})$, *i.e.,* when considering the standard relational algebra with the set semantics, the construction of Q in the previous proof reduces to the construction given by Paredaens [18]. More specifically, neither difference nor duplicate elimination are needed in this case to obtain BP-completeness, in accordance with the results in [18].

Example 14. Consider the relations S_3 and S_5 given in Figure 6. When viewed as $\mathcal{K}_{\mathbb{N}}$-relations, Theorem 1 guarantees the existence of a query Q in $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$ such that $Q(S_5) = S_3$. Although the query constructed in the proof of Theorem 1 is such a query, this query is by no means the unique (and most elegant) query with this property. Indeed, it is easily verified that $S_3 = ((\delta_{\mathbf{1}}(S_5) \cup \delta_{\mathbf{1}}(S_5)) \setminus S_5) \cup ((\delta_{\mathbf{1}}(S_5) \cup \delta_{\mathbf{1}}(S_5)) \setminus S_5)$. □

6. Conclusion

In view of the lack of expressive power of $\mathcal{RA}^+_{\mathcal{K}}$, we extended $\mathcal{RA}^+_{\mathcal{K}}$ with a difference operator, resulting in the query language $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$, constant annotation operators δ, resulting in the query language $\mathcal{RA}^+_{\mathcal{K}}(\delta)$, and both operators resulting in $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$. We proposed extended provenance semirings for $\mathcal{RA}^+_{\mathcal{K}}(\setminus)$ and $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$ and established crucial properties of the newly defined query languages, in particular the factorization property. This naturally extends previous work on the positive relational algebra. Finally, we initiated the study of BP-completeness of query languages on $\mathcal{K}$-relations. In particular, we showed that for some semirings $\mathcal{K}$, $\mathcal{RA}^+_{\mathcal{K}}$ is not BP-complete. Our main result is that $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$ is BP-complete on $\mathcal{K}$-relations for a general class of semirings $\mathcal{K}$. More specifically, $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$ is BP-complete for semirings that can be extended with a monus operator and that are finitely generated. This class of semirings covers most of the semirings considered in the database literature so far. We also showed that neither the difference nor duplicate elimination can be omitted while still retaining BP-completeness.

In future work, we plan to find an exact characterization of when two $\mathcal{K}$-relations are related by means of a query in $\mathcal{RA}^+_{\mathcal{K}}$ and establish the complexity of deciding this problem. Also, it is interesting to study the semantics of $\mathcal{RA}^+_{\mathcal{K}}(\setminus, \delta)$ for provenance models different than the why- and how-provenance. Finally, in the spirit of Codd, it is challenging to find a characterization of the completeness of $\mathcal{RA}^+_{\mathcal{K}}$ and extensions thereof in terms of first-order logic.

Acknowledgments

We would like to thank Leonid Libkin for helpful discussions, Jan Van den Bussche, Todd J. Green and Val Tannen for comments on a preliminary version of this paper.

References

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
[2] A. V. Aho and J. D. Ullman. Universality of data retrieval languages. In *POPL '79*, pages 110–119. ACM, 1979.
[3] K. Amer. Equationally complete classes of commutative monoids with monus. *Algebra Universalis*, 18(1):129–131, 1984.
[4] F. Bancilhon. On the completeness of query languages for relational data bases. In *MFCS '79*, volume 64 of *Lecture Notes in Computer Science*, pages 112–123. Springer, 1978.
[5] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. In *ICDT '01*, volume 1973 of *Lecture Notes in Computer Science*, pages 316–330. Springer, 2001.
[6] P. Buneman and W. C. Tan. Provenance in databases. In *SIGMOD '07*, pages 1171–1173. ACM, 2007.
[7] S. Burris and H. Sankappanavar. *A course in universal algebra*. Springer-Verlag, 1981.
[8] E. F. Codd. Relational completeness of data base sublanguages. *IBM Research Report RJ 987, San Jose, California*, 1972.
[9] Y. Cui, J. Widom, and J. L. Wiener. Tracing the lineage of view data in a warehousing environment. *ACM TODS*, 25(2):179–227, 2000.
[10] N. Fuhr and T. Rölleke. A probabilistic relational algebra for the integration of information retrieval and database systems. *ACM Trans. Inf. Syst.*, 15(1):32–66, 1997.
[11] T. J. Green. Containment of conjunctive queries on annotated relations. In *ICDT*, pages 296–309, 2009.
[12] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In *PODS '07*, pages 31–40. ACM, 2007.
[13] T. J. Green and V. Tannen. Models for incomplete and probabilistic information. *IEEE Data Eng. Bull.*, 29(1):17–24, 2006.
[14] M. Henriksen and J. R. Isbell. Lattice-ordered rings and function rings. *Pacific J. Math.*, 12:533–565, 1962.
[15] T. Imieliński and J. W. Lipski. Incomplete information in relational databases. *J. ACM*, 31(4):761–791, 1984.
[16] L. Libkin and L. Wong. Query languages for bags and aggregate functions. *J. Comput. Syst. Sci.*, 55(2):241–272, 1997.
[17] F. Montagna and V. Sebastiani. Equational fragments of systems for arithmetic. *Algebra Universalis*, 46(3):417–441, 2001.
[18] J. Paredaens. On the expressive power of the relational algebra. *Inf. Process. Lett.*, 7(2):107–111, 1978.
[19] E. Zimányi. Query evaluation in probabilistic relational databases. In *Selected papers from the international workshop on Uncertainty in databases and deductive systems*, pages 179–219. Elsevier, 1997.