Consistency Witnesses for Annotated Relations

Albert Atserias
Universitat Politècnica de Catalunya &
Centre de Recerca Matemàtica
Barcelona, Catalonia, Spain
atserias@cs.upc.edu

Phokion G. Kolaitis
UC Santa Cruz & IBM Research
Santa Cruz, California, USA
kolaitis@ucsc.edu

ABSTRACT

The study of local consistency vs. global consistency of database relations received considerable attention in the early days of relational database theory. In a recent paper, we investigated the notions of local consistency and global consistency for annotated relations, where the annotations come from a positive commutative monoid. One of the differences from the classical case is that the join of two consistent annotated relations need not always be a witness of their consistency. Here, we bring to center stage the notion of a consistency witness function for annotated relations, investigate the properties of consistency witness functions, and provide a new perspective to understanding the interplay between local and global consistency for annotated relations.

1 Introduction

During the past two decades, there has been a growing body of research on annotated databases, i.e., databases in which each fact is annotated with a value from some algebraic structure. This framework generalizes both standard relational databases, where the annotations are 1 (true) and 0 (false), and bag databases, where the annotations are non-negative integers denoting the multiplicity of a fact in the database. Much of the work in this area uses annotations from the universe of some fixed semiring $\mathbb{K} = (K, +, \times, 0, 1)$, where the addition operation $+$ is used to model "alternative" information (e.g., disjunction or existential quantification), while the multiplication operation $\times$ is used to model "joint" information (e.g., conjunction or universal quantification). For this reason, the term *semiring semantics* is often used to refer to the work in this area. Database provenance was the first extensively studied topic in this framework [8, 10, 4]. Subsequent studies focused on conjunctive query containment for annotated databases [7, 12], semiring semantics for first-order logic [6], and evaluation of Datalog programs under semiring semantics [11].

Since the early days of the relational database model, the study of consistency of relations has received significant attention [9, 3, 5]. By definition, a collection of relations $R_1, \ldots, R_m$ is *globally consistent* if there is a relation T such that the projection of T on the attributes of R_i is equal to R_i, for each $i = 1, \ldots, m$. We call such a relation T a *consistency witness for* $R_1, \ldots, R_m$. It is well known that if the collection $R_1, \ldots, R_m$ is globally consistent, then the join $R_1 \bowtie \cdots \bowtie R_m$ is a consistency witness for $R_1, \ldots, R_m$; in fact, it is the largest such consistency witness (see, e.g., [9]). As pointed out in [1], however, the state of affairs is different for bags, since there are two bags that are consistent but their join is not a consistency witness for them; moreover, no largest consistency witness for these bags exists.

In [2], we carried out an investigation of the consistency of annotated relations. Since the definition of consistency of annotated relations involves only the projection operation on relations and since projection is defined using only addition $+$, we considered annotated relations in which the annotations come from a monoid $\mathbb{K} = (K, +, 0)$. The main focus of that investigation was the interplay between local consistency and global consistency, that is, under what conditions a collection of pairwise consistent relations $R_1, \ldots, R_m$ is globally consistent. In particular, we identified a condition on monoids, called the *transportation property*, and showed that a positive monoid $\mathbb{K} = (K, +, 0)$ has the transportation property if and only if every acyclic hypergraph H has the local-to-global consistency property for $\mathbb{K}$-relations, which means that every pairwise consistent collection of $\mathbb{K}$-relations over H is globally consistent. This finding generalizes results about local vs. global consistency for standard relations in [3], as well as results about local vs. global consistency for bags in [1].

In this paper, we bring to front stage the notion of a *consistency witness function* on a positive monoid $\mathbb{K}$, that is to say, a function W that, given two $\mathbb{K}$-relations R and S, returns a $\mathbb{K}$-relation $W(R, S)$ that is a consistency witness for R and S, provided that R and S are consistent $\mathbb{K}$-relations. While the notion of a consistency witness function on $\mathbb{K}$ underlies much of the work in [2], it has not been studied in its own right thus far. Our goal is to make the case that this is a fundamental notion whose study is well deserved.

After presenting some basic properties of consistency witness functions on $\mathbb{K}$, we introduce the two notions of a c-*join-expression* and a *monotone* c-*join expression* for a consistency witness function on $\mathbb{K}$. These notions extend the notions of join expression and monotone join expressions for the standard join $\bowtie$ operation and for standard relations in [3, 5]. We then establish that the transportation property of a positive monoid $\mathbb{K}$ can be characterized in terms of properties of monotone c-join expressions. Furthermore, we argue that the notion a consistency witness function provides a new perspective to the proofs of the main results in [2]. We elaborate on this new perspective here and, along the way, we discuss methods for defining or constructing consistency witness functions for different types of monoids. Finally, we present some observations concerning the existence of "largest" consistency witness functions for annotated relations. In particular, we point out that a positive monoid being idempotent is a sufficient, but not necessary, condition for the existence of "largest" consistency witness functions for relations annotated with elements from that monoid.

2 Preliminaries

Monoids A *commutative monoid* is a structure $\mathbb{K} = (K, +, 0)$, where $+$ is a binary operation on the universe K of $\mathbb{K}$ that is associative, commutative, and has 0 as its neutral element, i.e., $p + 0 = p = 0 + p$ holds for all $p \in K$. A commutative monoid $\mathbb{K} = (K, +, 0)$ is *positive* if for all elements $p, q \in K$ with $p + q = 0$, we have that $p = 0$ and $q = 0$. From now on, we assume that all commutative monoids considered have at least two elements in their universe.

As an example, the structure $\mathbb{B} = (\{0, 1\}, \vee, 0)$ with disjunction $\vee$ as its operation and 0 (false) as its neutral element is a positive commutative monoid. Other examples of positive commutative monoids include the structure $\mathbb{N} = (Z^{\geq 0}, +, 0)$, and $\mathbb{R}^{\geq 0} = (R^{\geq 0}, +, 0)$, where $Z^{\geq 0}$ is the set of non-negative integers, $R^{\geq 0}$ is the set of non-negative real numbers, and $+$ is the standard addition operation. In contrast, the structure $\mathbb{Z} = (Z, +, 0)$, where Z is the set of integers, is a commutative monoid, but not a positive one. Two examples of positive commutative monoids of different flavor are the structures $\mathbb{T} = (R \cup \{\infty\}, \min, \infty)$ and $\mathbb{V} = ([0, 1], \max, 0)$, where R is the set of real numbers, and $\min$ and $\max$ are the standard minimum and maximum operations. Finally, if A is a set and $\mathcal{P}(A)$ is its powerset, then the structure $\mathbb{P}(A) = (\mathcal{P}(A), \cup, \emptyset)$ is a positive commutative monoid, where $\cup$ is the union operation on sets.

$\mathbb{K}$-relations and their marginals An *attribute* A is a symbol with an associated set $\mathrm{Dom}(A)$ as its *domain*. If X is a finite set of attributes, then we write $\mathrm{Tup}(X)$ for the set of X-*tuples*, i.e., $\mathrm{Tup}(X)$ is the set of functions that take each attribute $A \in X$ to an element of its domain $\mathrm{Dom}(A)$. Note that $\mathrm{Tup}(\emptyset)$ is non-empty as it contains the *empty tuple*, i.e., the unique function with empty domain. If $Y \subseteq X$ is a subset of attributes and t is an X-tuple, then the *projection of t on Y*, denoted by $t[Y]$, is the unique Y-tuple that agrees with t on Y. In particular, $t[\emptyset]$ is the empty tuple.

Let $\mathbb{K} = (K, +, 0)$ be a positive commutative monoid and let X be a finite set of attributes. A $\mathbb{K}$-*relation over* X is a function $R : \mathrm{Tup}(X) \to K$ that assigns a value $R(t)$ in K to every X-tuple t in $\mathrm{Tup}(X)$. We will often write $R(X)$ to indicate that R is a $\mathbb{K}$-relation over X, and we will refer to X as the set of attributes of R. These notions make sense even if X is the empty set of attributes, in which case a $\mathbb{K}$-relation over X is simply a single value from K that is assigned to the empty tuple. Clearly, the $\mathbb{B}$-relations are just the standard relations, while the $\mathbb{N}$-relations are the *bags* or *multisets*, i.e., each tuple has a non-negative integer associated with it that denotes the *multiplicity* of the tuple.

The *support* $\mathrm{Supp}(R)$ of a $\mathbb{K}$-relation $R(X)$ is the set of X-tuples t that are assigned non-zero value, i.e.,

$$\mathrm{Supp}(R) := \{t \in \mathrm{Tup}(X) : R(t) \neq 0\}. \qquad (1)$$

We will often write R' to denote $\mathrm{Supp}(R)$. Note that R' is a standard relation over X. A $\mathbb{K}$-relation is *finitely supported* if its support is a finite set. In this paper, all $\mathbb{K}$-relations considered will be finitely supported, and we omit the term; thus, from now on, a $\mathbb{K}$-relation is a finitely supported $\mathbb{K}$-relation. When R' is empty, we say that R is the empty $\mathbb{K}$-relation over X.

If $Y \subseteq X$, then the *marginal* $R[Y]$ *of* R *on* Y is the $\mathbb{K}$-relation over Y such that for every Y-tuple t, we have that

$$R[Y](t) := \sum_{\substack{r \in R': \\ r[Y]=t}} R(r). \qquad (2)$$

The value $R[Y](t)$ is the *marginal of* R *over* t. In what follows and for notational simplicity, we will often write $R(t)$ for the marginal of R over t, instead of $R[Y](t)$. It will be clear from the context (e.g., from the arity of the tuple t) if $R(t)$ is indeed the marginal of R over t (in which case t must be a Y-tuple) or $R(t)$ is the actual value of R on t as a mapping from $\mathrm{Tup}(X)$ to K (in which case t must be an X-tuple). Note that if R is a standard relation (i.e., R is a $\mathbb{B}$-relation), then the marginal $R[Y]$ is the projection of R on Y.

The proof of the next basic proposition follows easily from the definitions.

PROPOSITION 1. *Let $\mathbb{K}$ be a positive commutative monoid and let $R(X)$ be a $\mathbb{K}$-relation. Then the following hold:*

1. *For all $Y \subseteq X$, we have $R'[Y] = R[Y]'$.*
2. *For all $Z \subseteq Y \subseteq X$, we have $R[Y][Z] = R[Z]$.*

If X and Y are sets of attributes, then we write XY as shorthand for the union $X \cup Y$. Accordingly, if x is an X-tuple and y is a Y-tuple such that $x[X \cap Y] = y[X \cap Y]$, then we write xy to denote the XY-tuple that agrees with x on X and on y on Y. We say that x *joins with* y, and that y *joins with* x, to *produce* the tuple xy.

A *schema* is a sequence $X_1, \ldots, X_m$ of sets of attributes. A schema can also be identified with a hypergraph H having $X_1, \ldots, X_m$ as its hyperedges. We will use the terms *schema* and *hypergraph* interchangeably. A *collection of* $\mathbb{K}$*-relations* over such a schema is a sequence $R_1(X_1), \ldots, R_m(X_m)$ of $\mathbb{K}$-relations so that $R_i(X_i)$ is a $\mathbb{K}$-relation over X_i, for $i = 1, \ldots, m$.

3 Consistency and Consistency Witnesses

Let $\mathbb{K} = (K, +, 0)$ be a positive commutative monoid.

We say that two $\mathbb{K}$-relations $R(X)$ and $S(Y)$ are *consistent* if there is a $\mathbb{K}$-relation $T(XY)$ such that $T[X] = R$ and $T[Y] = S$. Such a $\mathbb{K}$-relation T is called a *consistency witness* for R and S.

A *consistency witness function on* $\mathbb{K}$ is a binary function W that takes as arguments two $\mathbb{K}$-relation $R(X)$ and $S(Y)$, and returns as value a $\mathbb{K}$-relation $W(R, S)$ over XY such that if R and S are consistent $\mathbb{K}$-relations, then $W(R, S)$ is a consistency witness for R and S. For example, the join $\bowtie$ of two standard relations is a consistency witness function on the Boolean monoid $\mathbb{B}$.

We say that a collection $R_1(X_1), \ldots, R_m(X_m)$ of $\mathbb{K}$-relations over a schema $X_1, \ldots, X_m$ is *globally consistent* if there is a $\mathbb{K}$-relation $T(X_1 \ldots X_m)$ such that $T[X_i] = R_i$, for i with $1 \leq i \ \leq m$. Such a $\mathbb{K}$-relation T is called a *consistency witness* for $R_1, \ldots, R_m$.

It is easy to see that if $R_1(X_1), \ldots, R_m(X_m)$ is a globally consistent collection of $\mathbb{K}$-relations, then these relations are pairwise consistent. Indeed, if T is a consistency witness for $R_1(X_1), \ldots, R_m(X_m)$, then for all i and j with $1 \leq i, j \leq m$, we have that the $\mathbb{K}$-relation $T[X_iX_j]$ is a consistency witness for R_i and R_j, because

$$R_i = T[X_i] = T[X_iX_j][X_i]$$
$$R_j = T[X_j] = T[X_iX_j][X_j]$$

where, in each case, the first equality follows from the definition of global consistency and the second equality follows from Proposition 1.

The converse is known to fail, even for standard relations, i.e., there are standard relations that are pairwise consistent but not globally consistent. The main result by Beeri et al. [3] characterizes the schemas for which the pairwise consistency of a collection of standard relations implies that they are globally consistent. Later on in this paper, we will see how this result extends to $\mathbb{K}$-relations over positive monoids satisfying a condition we call the *transportation property*.

We are interested in obtaining global consistency witnesses by using consistency witnesses for two relations. To this effect, we introduce certain syntactic expressions, which, under some additional hypotheses, will give rise to global consistency witnesses. In what follows, $\bowtie_{\rm c}$ is a binary function symbol, which will be interpreted by some consistency witness function.

Assume that $X_1, \ldots, X_m$ is a schema.

The collection of $\rm c$*-join expressions over* $X_1, \ldots, X_m$ is the smallest collection of strings that contains each X_i and has the property that if E_1 and E_2 are in the collection, then also the string $(E_1 \bowtie_{\rm c} E_2)$ is in the collection.

The collection of *sequential* $\rm c$*-join expressions over* $X_1, \ldots, X_m$ is the smallest collection of strings that contains each X_i and has the property that if E in the collection and X is one of the X_i's, then also the string $(E \bowtie_{\rm c} X)$ is in the collection.

A $\rm c$-join expression over $X_1, \ldots, X_m$ is called *read-once* if each X_i appears exactly once in the expression. We write $E[X_1, \ldots, X_m]$ to denote the read-once sequential $\rm c$-join-expression on $X_1, \ldots, X_m$ where the X_i appear in the indicated order; in the sequel, we refer to $E[X_1, \ldots, X_m]$ as the *read-once sequential* $\rm c$*-join-expression associated with the ordering* $X_1 \ldots, X_m$. In symbols, we have that $E[X_1, \ldots, X_m]$ is the $\rm c$-join expression

$$(\cdots((X_1 \bowtie_{\rm c} X_2) \bowtie_{\rm c} X_3) \bowtie_{\rm c} \cdots \bowtie_{\rm c} X_m).$$

Clearly, the string $((X_1 \bowtie_{\rm c} X_2) \bowtie_{\rm c} X_3)$ is a sequential $\rm c$-join-expression, while the string

$$((X_1 \bowtie_{\rm c} X_2) \bowtie_{\rm c} (X_3 \bowtie_{\rm c} X_4))$$

is a $\rm c$-join expression, but not a sequential one. Furthermore, both these strings are read-once $\rm c$-join expressions, while $((X_1 \bowtie_{\rm c} X_2) \bowtie_{\rm c} (X_3 \bowtie_{\rm c} X_1))$ is not. From now on we drop the outermost parentheses.

The notion of a $\rm c$-join expression is a syntactic one. We will now assign semantics to $\rm c$-join expressions.

Let $X_1, \ldots, X_m$ be a schema and let E be a $\rm c$-join-expression over $X_1, \ldots, X_m$. If W is a consistency witness function on $\mathbb{K}$ and $R_1(X_1), \ldots, R_m(X_m)$ is a collection of $\mathbb{K}$-relations, we write $E(W, R_1, \ldots, R_m)$ to denote the $\mathbb{K}$-relation over $X_1 \cdots X_m$ obtained by evaluating E when $\bowtie_{\rm c}$ is interpreted by W and each X_i is interpreted by R_i for $i = 1, \ldots, m$.

We say that E is *monotone with respect to* W *and* $R_1, \ldots, R_m$ if for every sub-expression $E_1 \bowtie_{\rm c} E_2$ of E, we have that the $\mathbb{K}$-relations $E_1(W, R_1, \ldots, R_m)$ and $E_2(W, R_1, \ldots, R_m)$ are consistent.

According to the next proposition, monotone c-join-expressions give rise to global consistency witnesses.

PROPOSITION 2. *Let E be a* c*-join expression over $X_1, \dots, X_m$, let W be a consistency witness function on $\mathbb{K}$, and let $R_1(X_1), \dots, R_m(X_m)$ be $\mathbb{K}$-relations. If E is monotone with respect to W and $R_1, \dots, R_m$, and every X_i occurs in E, then $E(W, R_1, \dots, R_m)$ is a global consistency witness for the $\mathbb{K}$-relations $R_1, \dots, R_m$.*

This proposition is proved by induction on the construction of c-join expressions.

The base case is trivial, since in this case E is X_i for some i with $1 \leq i \leq n$, hence $E(W, R_i) = R_i$, which is a consistency witness for R_i.

For the inductive step, assume that E is $E_1 \bowtie_{\mathrm{c}} E_2$, where E_1 and E_2 are c-join expressions for which the inductive hypothesis holds. To simplify the notation, let us put $\mathbf{R} = (R_1, \dots, R_m)$; furthermore, we put $\mathbf{R}_1 = (R_i : i \in I_1)$ and $\mathbf{R}_2 = (R_i : i \in I_2)$, where I_1 and I_2 are the sets of indices i such that X_i occurs in E_1 and in E_2, respectively. In this case, we have that $E(W, \mathbf{R}) = W(E_1(W, \mathbf{R}_1), E_2(W, \mathbf{R}_2))$.

Since E is monotone with respect to W and $\mathbf{R}$, we have that the $\mathbb{K}$-relations $E_1(W, \mathbf{R}_1)$ and $E_2(W, \mathbf{R}_2)$ are consistent, hence $W(E_1(W, \mathbf{R}_1), E_2(W, \mathbf{R}_2))$ is a consistency witness for $E_1(W, \mathbf{R}_1)$ and $E_2(W, \mathbf{R}_2)$. We must show that $W(E_1(W, \mathbf{R}_1), E_2(W, \mathbf{R}_2))[X_i] = R_i$ holds, for every i such that X_i occurs in E. Consider such an X_i. Since X_i occurs in E, it must occur in at least one of E_1 and E_2. Let's assume that X_i occurs in E_1; the case in which it occurs in E_2 is entirely similar. If Y is the set of attributes of $E_1(W, \mathbf{R}_1)$, then $X_i \subseteq Y$. Furthermore, the property of an expression being monotone with respect to a witness function and a collection of relations is inherited by its subexpressions, so E_1 is monotone with respect to W and $\mathbf{R}_1$. By induction hypothesis, $E_1(W, \mathbf{R}_1)$ is a global consistency witness of all relations R_j occurring in it, hence

$$E_1(W, \mathbf{R}_1)[X_i] = R_i. \tag{3}$$

Also, since $W(E_1(W, \mathbf{R}_1), E_2(W, \mathbf{R}_2))$ is a consistency witness for $E_1(W, \mathbf{R}_1)$ and $E_2(W, \mathbf{R}_2)$, we have that

$$W(E_1(W, \mathbf{R}_1), E_2(W, \mathbf{R}_2))[Y] = E_1(W, \mathbf{R}_1). \tag{4}$$

By putting everything together, we have that

$$\begin{aligned} &W(E_1(W, \mathbf{R}_1), E_2(W, \mathbf{R}_2))[X_i] \\ &= W(E_1(W, \mathbf{R}_1), E_2(W, \mathbf{R}_2))[Y][X_i] \\ &= E_1(W, \mathbf{R}_1)[X_i] \\ &= R_i, \end{aligned}$$

where in the first equality we used Proposition 1 and the fact that $X_i \subseteq Y$, in the second we used (4), and the third is (3). This completes the proof of Proposition 2.

4 The Transportation Property

We consider several different properties of monoids and establish that they are equivalent to each other.

Let $\mathbb{K} = (K, +, 0)$ be a positive commutative monoid.

If m and n are positive integers, we say that $\mathbb{K}$ has the $m \times n$ *transportation property* if for every column m-vector $b = (b_1, \dots, b_m) \in K^m$ with entries in K and every row n-vector $c = (c_1, \dots, c_n) \in K^n$ with entries in K such that $b_1 + \cdots + b_m = c_1 + \cdots + c_n$ holds, there is an $m \times n$ matrix $D = (d_{ij} : i \in [m], j \in [n]) \in K^{m \times n}$ with entries in K whose rows sum to b and whose columns sum to c, i.e., $d_{i1} + \cdots + d_{im} = b_i$ for all $i \in [m]$ and $d_{1j} + \cdots + d_{mj} = c_j$ for all $j \in [n]$.

We say that $\mathbb{K}$ has the *transportation property* if $\mathbb{K}$ has the $m \times n$ transportation property for every pair (m, n) of positive integers.

We now consider a number of properties of monoids that involve $\mathbb{K}$-relations.

Two $\mathbb{K}$-relations $R(X)$ and $S(Y)$ are *inner consistent* if $R[X \cap Y] = S[X \cap Y]$. Using Proposition 1, it is easy to verify that if R and S are consistent $\mathbb{K}$-relations, then they are also inner consistent. The converse, however, is not true for all positive commutative monoids. We single out the ones for which inner consistency implies consistency (consequently, for such monoids, these two notions are equivalent).

We say that $\mathbb{K}$ has the *inner consistency property* if whenever two $\mathbb{K}$-relations are inner consistent, then they are also consistent.

We say that a schema $X_1, \dots, X_m$ has the *local-to-global consistency property for $\mathbb{K}$-relations* if every collection $R_1(X_1), \dots, R_m(X_m)$ of pairwise consistent $\mathbb{K}$-relations is also globally consistent.

Let E be a c-join-expression over $X_1, \dots, X_m$. We say that E is *monotone on* $\mathbb{K}$ if E is monotone with respect to every consistency witness function W on $\mathbb{K}$ and every collection $R_1(X_1), \dots, R_m(X_m)$ of pairwise consistent $\mathbb{K}$-relations.

Finally, we say that a schema $X_1, \dots, X_m$ *admits a monotone* c*-join expression on* $\mathbb{K}$ if there is a c-join-expression E over $X_1, \dots, X_m$ that is monotone on $\mathbb{K}$ and, furthermore, every X_i occurs in E.

THEOREM 1. *The following statements are equivalent for a positive monoid $\mathbb{K}$:*

1. *$\mathbb{K}$ has the 2×2 transportation property.*
2. *$\mathbb{K}$ has the transportation property.*
3. *$\mathbb{K}$ has the inner consistency property.*
4. *Every acyclic hypergraph admits a monotone read-once sequential* c*-join-expression on $\mathbb{K}$.*
5. *Every acyclic hypergraph admits a monotone read-once* c*-join-expression on $\mathbb{K}$.*

6. *Every acyclic hypergraph admits a monotone* c*-join-expression on* $\mathbb{K}$*.*
7. *Every acyclic hypergraph has the local-to-global consistency property for* $\mathbb{K}$*-relations.*

The proofs of the implications $(1) \Rightarrow (2)$ and $(2) \Rightarrow (3)$ are given in [2]. A new perspective on these proofs will be presented in Section 6. Here, we sketch the proofs of the remaining implications in a round-robin fashion.

We begin with the implication $(3) \Rightarrow (4)$. As shown in Beeri et al. [3], if H is an acyclic hypergraph, then H has the *running intersection* property, which means that there is an ordering $X_1, \ldots, X_m$ of the hyperedges of H so that for every $j \leq m$, there is some $i \leq j-1$ such that $(X_1 \cup \cdots \cup X_{j-1}) \cap X_j \subseteq X_i$. Let $E[X_1, \ldots, X_m]$ be the read-once sequential c-join-expression associated with this ordering, i.e., $E[X_1, \ldots, X_m]$ is

$$(\cdots((X_1 \bowtie_{\mathrm{c}} X_2) \bowtie_{\mathrm{c}} X_3) \bowtie_{\mathrm{c}} \cdots \bowtie_{\mathrm{c}} X_m).$$

Using the inner consistency property of $\mathbb{K}$, it is not hard to show that $E[X_1, \ldots, X_m]$ is monotone on $\mathbb{K}$. The implications $(4) \Rightarrow (5)$ and $(5) \Rightarrow (6)$ are trivial. The implication $(6) \Rightarrow (7)$ uses Proposition 2 and, of course, the definitions.

Finally, we prove $(7) \Rightarrow (1)$. We are given a 2×2 instance of the transportation problem on $\mathbb{K}$: four elements $b_1, b_2, c_1, c_2 \in K$ such that $b_1 + b_2 = c_1 + c_2$. Consider the following three $\mathbb{K}$-relations where $e = b_1 + b_2 = c_1 + c_2$:

AB	BC	CD
$1\,0 : b_1$	$0\,0 : e$	$0\,1 : c_1$
$2\,0 : b_2$	$1\,1 : e$	$0\,2 : c_2$
$1\,1 : c_1$		$1\,1 : b_1$
$2\,1 : c_2$		$1\,2 : b_2$

It is easy to see that these are pairwise consistent, and the schema is acyclic as it is the path of length three. By (7) the three $\mathbb{K}$-relations are also globally consistent. Let $W(ABCD)$ be a witness of global consistency. Setting $d_{ij} = W(i00j)$ or $d_{ij} = W(j11i)$ we get a solution to the 2×2 instance, which completes the proof.

Beeri et al. showed that a hypergraph H is acyclic if and only if H has the local-to-global consistency property for standard relations (i.e., for $\mathbb{B}$-relations, where $\mathbb{B}$ is the Boolean monoid). In [2], we showed that if $\mathbb{K}$ is an arbitrary positive monoid and H is a hypergraph that has the local-to-global consistency property for $\mathbb{K}$-relations, then H must be acyclic. We also showed that there are positive commutative monoids $\mathbb{K}$ and acyclic schemas H that do *not* have the local-to-global consistency property for $\mathbb{K}$-relations. Thus, acyclicity is a necessary, but not sufficient, condition for H to have the local-to-global consistency property for $\mathbb{K}$-relations. Theorem 1, however, implies that acyclicity is both necessary and sufficient, provided $\mathbb{K}$ has the transportation property. Thus, we have the following generalization of the main result in Beeri et al. [3].

Theorem 2. *Assume that* $\mathbb{K}$ *is a positive commutative monoid that has the transportation property. For every hypergraph* H*, the following statements are equivalent:*

1. H *is acyclic.*
2. H *admits an ordering* $X_1, \ldots, X_m$ *of its hyperedges so that the sequential* c*-join expression associated with* $X_1, \ldots, X_m$ *is monotone on* $\mathbb{K}$*.*
3. H *admits a monotone* c*-join-expression on* $\mathbb{K}$*.*
4. H *has the local-to-global consistency property for* $\mathbb{K}$*-relations.*

Naturally, in Theorem 2 we can also add as equivalent statements that H admits a sequential monotone c-join-expression on $\mathbb{K}$, as well as a read-once sequential monotone c-join-expression on $\mathbb{K}$. Recall that this last condition is equivalent to the statement that there is an ordering $X_1, \ldots, X_m$ of the hyperedges of H so that the sequential c-join expression $E[X_1, \ldots, X_m]$ associated with $X_1, \ldots, X_m$ is monotone on $\mathbb{K}$.

5 Defining Consistency Witnesses

By definition, every consistency witness function for a positive commutative monoid $\mathbb{K}$ produces a consistency witness $W = W(R, S)$, given two consistent $\mathbb{K}$-relations $R(X)$ and $S(Y)$. But how can such a consistency witness function be defined? Are there general ways of constructing a consistency witness function?

For several specific monoids of interest, the consistency witness can be found via an explicit expression or via a procedural method. For example, for the Boolean monoid $\mathbb{B}$, the standard join $R \bowtie S$ of standard relations is an explicit consistency witness function. More generally, if $\mathbb{K} = (K, \vee, 0)$ is the join semilattice of a bounded distributive lattice $(K, \vee, \wedge, 0, 1)$ (the same way the Boolean monoid $\mathbb{B}$ is the join semilattice of the 2-element Boolean algebra), then setting

$$W(t) = R(t[X]) \wedge S(t[Y]) \tag{5}$$

for every XY-tuple t gives an explicit expression that defines a consistency witness function for every two consistent $\mathbb{K}$-relations $R(X)$ and $S(Y)$.

Similarly, if $\mathbb{K} = (K, +, 0)$ is the additive monoid of a semifield $(K, +, \times, /, 0, 1)$, (the same way the positive monoid $\mathbb{R}^{\geq 0}$ of non-negative reals with addition is the additive monoid of the semifield of non-negative real numbers with addition and multiplication), then an explicit expression for a consistency witness is given by setting

$$W(t) = R(t[X]) \times S(t[Y])/D(t) \tag{6}$$

where $D(t) = R(t[X \cap Y]) = S(t[X \cap Y])$ with the convention that $0/0 = 0$. Note that the equality in the definition of $D(t)$ follows from the assumption that R and S are consistent; indeed, if U witnesses their consistency, then

$$R[X \cap Y] = U[X][X \cap Y] = U[Y][X \cap Y] = S[X \cap Y],$$

where the middle equation follows from Proposition 1.

The expressions in (5) and (6) are called respectively the *standard join* of the distributive lattice, which is denoted by $R \bowtie_{\mathbb{K}} S$, and the *Vorobe'v join* of the semifield, which is denoted by $R \bowtie_{\mathrm{V}\mathbb{K}} S$.

When it comes to the bag monoid $\mathbb{N} = (N, +, 0)$, it turns out that the standard join of bags is *not* a valid consistent witness function. For example, the two bags $R(X) = \{a{:}1, b{:}1\}$ and $S(Y) = \{c{:}1, d{:}1\}$ are consistent via the witness $\{ac{:}1, bd{:}1\}$ or $\{ad{:}1, bc{:}1\}$, but their bag join is the bag $\{ac{:}1, ad{:}1, bc{:}1, bd{:}1\}$, which projects to $\{a{:}2, b{:}2\}$ on X and to $\{c{:}2, d{:}2\}$ on Y, thus it is not a witness of their consistency. Nonetheless, the bag monoid does admit an explicit consistency witness function, which can be defined via a procedure called the Northwest Corner Method. As explained in [2], the inspiration for this procedure came from linear programming, simplifying an earlier method from [1]. In Section 7 we provide an alternative perspective to it.

We refer the reader to Section 5 of [2] for an ample discussion of specific monoids and classes of monoids for which a consistency witness can be explicitly defined by an expression or by a procedural method, such as the Northwest Corner Method.

In the next section, we discuss a more general problem, which is implicit in the validity of the implications (1) ⇒ (2) ⇒ (3) of Theorem 1. The problem can be stated as follows: How can the transportation property alone be used to construct consistency witnesses in full generality? First we discuss how a direct interpretation of the proof of the implication (2) ⇒ (3) in Theorem 1 gives a way to construct consistency witnesses by solving explicit but typically large systems of equations over the monoid. Then we argue that the proof of the implication (1) ⇒ (2) in Theorem 1 indeed gives a way to construct witnesses from just solving 2×2 systems.

This is a rather remarkable phenomenon that enables the construction of consistency witnesses by repeatedly solving many but tiny 2×2 systems of equations over the monoid. This phenomenon is akin to the fact that the standard join of standard relations can be computed very efficiently (in terms of the output size) by scanning the pairs of tuples in the two relations in a carefully chosen order. As we will see, in the general case of positive commutative monoids with the transportation property, it suffices to scan not pairs of tuples (i.e., 1×1 systems) but *pairs of pairs of tuples* (i.e., 2×2 systems), also in some suitable order.

We begin our discussion by recalling the aforementioned standard and efficient method for computing joins of standard relations.

6 From 2 × 2 Systems to Witnesses

For relational databases, the Sort-Merge Join algorithm is a well-known method to compute the join of two relations $R(X)$ and $S(Y)$; e.g., see Section 12.5.2 in [13]. The algorithm works as follows.

First sort the tuples in R and S in the two relations lexicographically by the entries of the tuples on the common attributes $Z = X \cap Y$, i.e., sort all tuples $r \in R$ by $r[Z]$ and sort all tuples $s \in S$ by $s[Z]$. Then, scan the two sorted lists in parallel to find a tuple $t \in \mathrm{Tup}(Z)$ on the common attributes that appears in both lists. For each such t found, scan all pairs of tuples $r \in R$ and $s \in S$ such that $r[Z] = t$ and $s[Z] = t$, produce the join tuple rs in the output $W(XY)$, and proceed to the next common t in the sorted lists. Since the join of two consistent standard relations is a witness of their consistency, this algorithm computes a consistency witness function for the Boolean monoid $\mathbb{B}$.

When the positive monoid $\mathbb{K}$ has the transportation property, there is a natural analogue of the Sort-Merge Join algorithm that produces consistency witnesses for consistent $\mathbb{K}$-relations $R(X)$ and $S(Y)$. Again, first sort all tuples $r \in R'$ and all tuples $s \in S'$ in the supports R' and S' of R and S lexicographically by the entries of the tuples on the common attributes $Z = X \cap Y$. Then, scan the sorted lists to find the first tuple $t \in \mathrm{Tup}(Z)$ that appears in both lists. For such t, form a system of equations over $\mathbb{K}$. For each $r \in R'$ and $s \in S'$ such that $r[Z] = s[Z] = t$, the system has one variable $x_{r,s;t}$. The system has equations

$$\sum_{\substack{s \in S': \\ s[Z]=t}} x_{r,s;t} = b_r \quad \text{and} \quad \sum_{\substack{r \in R': \\ r[Z]=t}} x_{r,s;t} = c_s$$

for each $r \in R'$ with $r[Z] = t$ and $b_r = R(r)$ in the first equation, and each $s \in S'$ with $s[Z] = t$ and $c_s = S(s)$ in the second equation.

Now note that by the assumption that $R(X)$ and $S(Y)$ are consistent, we have $\sum_r b_r = \sum_s c_s$. By the transportation property of $\mathbb{K}$, the system has a solution in $\mathbb{K}$, say by setting $x_{r,s;t}$ to $a_{r,s;t}$. Finally, use this solution to produce the annotated tuple $rs{:}a_{r,s;t}$ in the output $W(XY)$ for each considered r and s, and proceed to the next common t in the sorted lists. The fact that the resulting $\mathbb{K}$-relation $W(XY)$ is a consistency witness for R and S is an immediate consequence of the definitions and the way the system of equations was set up. This construction is also what goes behind the scenes in the proof of the implication (2) ⇒ (3) in Theorem 1. We refer the reader to [2] for more details on this proof.

An important point about the Sort-Merge Join algorithm of the previous paragraph is that it involves solving systems of equations of many different sizes, and often very big ones. Concretely, if for a tuple t that appears in both lists we have m_t tuples $r \in R'$ such $r[Z] = t$ and n_t tuples $s \in S'$ such that $s[Z] = t$, then the system associated to tuple t has $m_t \times n_t$ variables and $m_t + n_t$ equations. Since m_t and n_t could in general be quite big, solving each such system for each tuple t individually could be computationally expensive. This should be compared with the explicit and usually efficiently computable expressions of Equation (5) for the standard join of a distributive lattice, and Equation (6) for the Vorobe'v join of a semifield. In contrast to these explicit expressions, if all we know about the monoid is that it has the transportation property, then no such explicit expression may be available; thus, it looks like we are stuck with the daunting task of solving potentially huge $m_t \times n_t$ systems of equations for each t.

Or are we?

Interestingly, the implication (1) ⇒ (2) in Theorem 1 asserts that the 2 × 2 transportation property *alone* already implies the $m \times n$ transportation property for every positive integers m and n. At least in principle, this means that in order to solve the $m_t \times n_t$ systems of each t it should suffice to solve perhaps many but tiny 2 × 2 systems. In the rest of this section we explain how the proof of the implication (1) ⇒ (2) in Theorem 1 can be leveraged to reduce the task for solving the $m_t \times n_t$ systems within the context of the Sort-Merge Join algorithm to that of solving many but tiny 2 × 2 systems.

To discuss this, let us first examine one possible implementation of the inner loop in the Sort-Merge Join algorithm for standard relations. The method we suggest below is almost certainly not what would be implemented in practice because, for practical implementations, iterative methods are preferred over recursive ones. However, it is conceptually useful to explain the method as a recursive algorithm to see how it generalizes to the case of $\mathbb{K}$-relations over monoids that have the transportation property.

Within the Sort-Merge Join algorithm for standard relations, let's say we are in the situation where we have detected a tuple $t \in \mathrm{Tup}(Z)$ that appears in both sorted lists of the tuples of $R(X)$ and $S(Y)$. The subroutine that we are about to describe produces all join-tuples rs for $r \in R$ and $s \in S$ such that $r[Z] = s[Z] = t$.

Let the sorted lists of such tuples be $r_1, \dots, r_m$ and $s_1, \dots, s_n$, respectively. If $m = 1$, then we output the join tuples $r_1 s_j$ for $j = 1, \dots, n$ and we are done. Symmetrically, if $n = 1$, then we output $r_i s_1$ for $i = 1, \dots, m$ and we are done again. Suppose then that $m \geq 2$ and $n \geq 2$. If $m > n$, then we split the problem into a base case with the singleton list r_m and a recursive case with the reduced list $r_1, \dots, r_{m-1}$. In both cases the other list remains $s_1, \dots, s_n$. Symmetrically, if $m < n$, then we split the problem into a base case with the singleton list s_n, and a recursive case with the reduced list $s_1, \dots, s_{n-1}$. Again, in both cases the other list remains $r_1, \dots, r_m$. In case $m = n$, we just break ties arbitrarily and go with one of the two. Since $m \geq 2$ and $n \geq 2$, the recursive calls made in this subroutine call always make progress in reducing the sizes of the lists and we end up producing all pairs $r_i s_j$, as required.

What we need to answer now is why we cannot just do the same for the variant of the Sort-Merge Join algorithm for consistent $\mathbb{K}$-relations. The base cases $m = 1$ and $n = 1$ can certainly be handled the same way, using the annotations $a_{r_1,s_j;t} = S(s_j)$ for the tuples $r_1 s_j$ in the case $m = 1$, and the annotations $a_{r_i,s_1;t} = R(r_i)$ for the tuples $r_i s_1$ in the case $n = 1$. The problem is with the subroutine call in the inductive case $m \geq 2$ and $n \geq 2$: the recursive call with the reduced list does not interact at all with the call with the singleton list, so it is hard to believe that the two calls will magically produce a solution $a_{r_i,s_j;t}$ that satisfies the equations of the consistency requirement. These equations impose global conditions that involve the full lists $r_1, \dots, r_m$ and $s_1, \dots, s_n$; therefore, they require some kind of coordination between calls. It is here where it is useful to upgrade the kind of processing that the algorithm does from handling pairs of tuples to handling pairs of pairs of tuples (i.e., 2 × 2 systems). Let us see how to do this.

As a reminder, it is useful to keep in mind the following graphical representation of the system of equations that we need to satisfy:

$$\begin{array}{ccccccc} x_{1,1} & + & \cdots & + & x_{1,n} & = & b_1 \\ + & & & & + & & \\ \vdots & & \ddots & & \vdots & & \\ + & & & & + & & \\ x_{m,1} & + & \cdots & + & x_{m,n} & = & b_n \\ \shortparallel & & & & \shortparallel & & \\ c_1 & & & & c_n & & \end{array} \tag{7}$$

where for simplicity we wrote $x_{i,j}$ instead of $x_{r_i,s_j;t}$ and b_i and c_j instead of $R(r_i)$ and $S(s_j)$.

The solution to the problem of non-interacting calls can be discovered by examining how we would manually handle the next limiting cases after the base cases. Let's say $m = 2$, so the first list of tuples is r_1, r_2 and the column vector in the right-hand side of the system (7) is b_1, b_2, and the second list of tuples is $s_1, \dots, s_n$ with $n \geq 2$. If we were able to solve any 2×2 instance of the transportation problem, then we could split the problem of solving the system (7) in the special case $m = 2$ as follows. First we solve the 2×2 system given by the equations

$$\begin{array}{ccccc} y_1 & + & x_{1,n} & = & b_1 \\ + & & + & & \\ y_2 & + & x_{2,n} & = & b_2 \\ \shortparallel & & \shortparallel & & \\ c & & c_n & & \end{array}$$

where y_1, y_2 are two new variables and $c = c_1 + \cdots + c_{n-1}$. Observe that $c + c_n = b_1 + b_2$, as required by the transportation property. Once this is solved, we go on recursively to solve the $2 \times (n-1)$ system given by the equations

$$\begin{array}{ccccccc} x_{1,1} & + & \cdots & + & x_{1,n-1} & = & y_1 \\ + & & & & + & & \\ x_{2,1} & + & \cdots & + & x_{2,n-1} & = & y_2 \\ \shortparallel & & & & \shortparallel & & \\ c_1 & & & & c_{n-1} & & \end{array}$$

Observe that $c_1 + \cdots + c_{n-1} = c = y_1 + y_2$, as required by the transportation property. Note also how the part y_1, y_2 of the solution to the first system is *used to define* the right-hand side of the second system, so the two calls of the subroutine *now do interact*. A simple inspection shows that the concatenation of the solutions of the two systems gives a solution to the global $m \times n$ system (7) in the special case $m = 2$.

This analysis takes care of the case $m = 2$ and $n \geq 2$. To take care of the case $m \geq 2$ and $n = 2$, we proceed symmetrically exchanging rows and columns. Finally, for the case $m \geq 3$ and $n \geq 3$, we can use the cases $2 \times n$ and $m \times 2$ that we just discussed as base cases. We split an $m \times n$ system as in (7) into a $2 \times n$ system and an $(m-1) \times n$ system if $m > n$, or into an $m \times 2$ system and an $m \times (n-1)$ system if $m < n$, breaking ties arbitrarily if $m = n$. This completes the description and the analysis of the recursive algorithm. The inductive argument that proves its correctness is also what goes behind the scenes in the proof of the implication (1) ⇒ (2) in Theorem 1, as presented in [2].

The bottom line of this section is that the Sort-Merge Join algorithm for computing joins of standard relations, and hence consistency witnesses of standard relations, nicely generalizes to an algorithm for computing consistency witnesses of two given consistent $\mathbb{K}$-relations from just knowing how to solve many but explicit and tiny 2×2 instances of the transportation problem.

7 Solving 2×2 Systems in Specific Cases

In view of the analysis of the previous section, it is now natural to revisit the question of solving 2×2 systems for specific monoids. In this section, we revisit the standard join and the Vorobe'v join in Section 5 from the perspective of 2×2 systems. We also give an explicit solution to the 2×2 systems for monoids for which such systems are solvable using the Northwest Corner Method, also mentioned in Section 5. As stated there, the most natural example of this last case is the bag monoid $\mathbb{N}$. By unfolding the recursive algorithm of the previous section, the explicit solution we give in this section gives an alternative and computationally more explicit definition of the Northwest Corner Method, as compared to how it was presented in [2].

Let $\mathbb{K} = (K, +, 0)$ be a positive commutative monoid. We are given b_1, b_2, c_1, c_2 such that $b_1 + b_2 = c_1 + c_2$. We want to solve the following system:

$$\begin{array}{ccccc} x_{11} & + & x_{12} & = & b_1 \\ + & & + & & \\ x_{21} & + & x_{22} & = & b_2 \\ \shortparallel & & \shortparallel & & \\ c_1 & & c_2 & & \end{array}$$

We may assume that all b_1, b_2, c_1, c_2 are different from 0 as otherwise we can set both variables of the corresponding row or column equation to 0 and reduce the system to a single trivially satisfiable equation.

Let $e = b_1 + b_2 = c_1 + c_2$ and note $e \neq 0$ by positivity.

If $\mathbb{K} = (K, \vee, 0)$ is the join semilattice of a bounded distributive lattice $(K, \vee, \wedge, 0, 1)$, then setting $x_{ij} = b_i \wedge c_j$ for $i, j = 1, 2$ gives a solution. Indeed, $x_{i1} \vee x_{i2} = (b_i \wedge c_1) \vee (b_i \wedge c_2) = b_i \wedge (c_1 \vee c_2) = b_i \wedge e = b_i \wedge (b_1 \vee b_2) = b_i$. An entirely symmetric argument gives $x_{1j} \vee x_{2j} = c_j$. Examples include the Boolean monoid, the power set monoid, and many others.

If $\mathbb{K} = (K, +, 0)$ is the additive monoid of a semifield $(K, +, \times, /, 0, 1)$, then setting $x_{ij} = (b_i \times c_j)/e$ for $i, j = 1, 2$ gives a solution. Indeed, $x_{i1} + x_{i2} = (b_i \times c_1)/e + (b_i \times c_2)/e = b_i \times (c_1 + c_2)/e = b_i$. Similarly, $x_{1j} + x_{2j} = c_j$. Examples include the nonnegative reals with addition, tropical monoids such as $(R \cup \{-\infty\}, \max, -\infty)$ and many others.

Finally we come to the bag monoid $\mathbb{N} = (Z^{\geq 0}, +, 0)$ and those positive monoids whose instances of the transportation problem can be solved by the Northwest Corner Method. We need some preliminary definitions.

Every positive commutative monoid $\mathbb{K} = (K, +, 0)$ is *canonically preordered* by the binary relation $x \leq y$ defined to hold between two elements $x, y \in K$ if there exists an element $z \in K$ such that $x + z = y$. If for any every two elements $x, y \in K$ we have $x \leq y$ or $y \leq x$ (or both), then we say that this preorder is *total* and that the monoid is *totally canonically preordered*. In such a case, the operation $\min(x, y)$, which returns x if $x \leq y$ and y otherwise, satisfies the inequalities $\min(x, y) \leq x$ and $\min(x, y) \leq y$. Similarly, the operation $\max(x, y)$, which returns y if $x \leq y$ and x otherwise, satisfies the inequalities $x \leq \max(x, y)$ and $y \leq \max(x, y)$. We say that $\mathbb{K}$ is *weakly cancellative* if for every x, y, z, we have that $x+y = x+z$ implies that $y = z$ or $y = 0$ or $z = 0$. When a monoid is weakly cancellative, it is natural to define an operation $x \dot{-} y$ on pairs x, y. Concretely, if $x \not\leq y$, then we set $x \dot{-} y = 0$, and if $x \leq y$ via $x+z = y$, then we set $x \dot{-} y = z$ if $x \neq y$ and $x \dot{-} y = 0$ if $x = y$. By weak cancellativity, $x \dot{-} y$ is well defined because if both $x+z = y$ and $x+z' = y$ hold, then $x+z = x+z'$ so by weak cancellativity we have $z = z'$, or $z = 0$ in which case $x = y$, or $z' = 0$ in which case again $x = y$. This operation has the property that if $x \leq y$, then $x + (y \dot{-} x) = y$.

Suppose now that $\mathbb{K}$ is totally canonically preordered and weakly cancellative. The typical example is the bag monoid $\mathbb{N}$, for which $\min(x, y)$ and $\max(x, y)$ are the minimum and the maximum operations, and $\dot{-}$ is the subtraction operation truncated to 0. In this case, a solution is given by the Northwest Corner Method, which in the 2×2 case reduces to the following explicit assignment (recall that $e = b_1 + b_2 = c_1 + c_2$ and b_1, b_2, c_1, c_2 are different from 0):

$$\begin{aligned} x_{11} &= \min(b_1, c_1) \\ x_{12} &= b_1 \dot{-} x_{11} \\ x_{21} &= c_1 \dot{-} x_{11} \\ x_{22} &= e \dot{-} \max(b_1, c_1) \end{aligned}$$

To see that this system satisfies the 2×2 system first observe that $x_{11} \leq b_1$ and $x_{11} \leq c_1$, so $x_{11} + x_{12} = x_{11} + (b_1 \dot{-} x_{11}) = b_1$ and $x_{11} + x_{21} = x_{11} + (c_1 \dot{-} x_{11}) = c_1$. This already shows that half of the equations of the 2×2 system are satisfied. For the remaining two equations, first we claim that

$$\begin{array}{ll} b_2 = (c_1 \dot{-} b_1) + c_2 & \text{if } b_1 \leq c_1 \\ c_2 = (b_1 \dot{-} c_1) + b_2 & \text{if } b_1 \not\leq c_1. \end{array} \qquad (8)$$

Indeed, if $b_1 \leq c_1$ then $c_1 = b_1 + (c_1 \dot{-} b_1)$, so we have $b_1 + b_2 = c_1 + c_2 = b_1 + (c_1 \dot{-} b_1) + c_2$. The first equality in (8) then follows from weak cancellativity because $b_2 \neq 0$ and $c_2 \neq 0$, and therefore also $(c_1 \dot{-} b_1) + c_2 \neq 0$ by positivity. Similarly, if $b_1 \not\leq c_1$, then we have $c_1 \leq b_1$ because the preorder is total, so $b_1 = c_1 + (b_1 \dot{-} c_1)$ and we have $c_1 + c_2 = b_1 + b_2 = c_1 + (b_1 \dot{-} c_1) + b_2$. The second equality in (8) follows then again by weak cancellativity and positivity. Now we use (8) to show, by cases, that the remaining two equations of the 2×2 system are satisfied.

If $b_1 \leq c_1$, then $x_{12} = b_1 \dot{-} b_1 = 0$ and $x_{21} = c_1 \dot{-} b_1$, as well as $x_{22} = e \dot{-} c_1 = c_2$ because $c_1 + c_2 = e$ and therefore $c_1 \leq e$. This shows that $x_{12} + x_{22} = c_2$ and $x_{21} + x_{22} = b_2$ by (8). Similarly, if $b_1 \not\leq c_1$, then $x_{21} = c_1 \dot{-} c_1 = 0$ and $x_{12} = b_1 \dot{-} c_1$, as well as $x_{22} = e \dot{-} b_1 = b_2$ because $b_1 + b_2 = e$ and therefore $b_1 \leq e$. This shows that $x_{21} + x_{22} = b_2$ and $x_{12} + x_{22} = c_2$ by (8).

8 Largest Consistency Witnesses

A key fact about standard relations is that if $R(X)$ and $S(Y)$ are two consistent standard relations, then there is a consistency witness $W(XY)$ for $R(X)$ and $S(Y)$ that is *largest* in the sense that every other consistency witness $U(XY)$ for $R(X)$ and $S(Y)$ is included in it, i.e., $U \subseteq W$ holds. This follows from the basic fact that if $W_1(XY)$ and $W_2(XY)$ are consistency witnesses for the standard relations $R(X)$ and $S(Y)$, then their set-theoretic union $W_1 \cup W_2$ is also a consistency witness for $R(X)$ and $S(Y)$. Therefore, the union of all consistency witnesses for $R(X)$ and $S(Y)$ is the largest consistency witness for them (and it actually coincides with the standard join $R \bowtie S$).

Assume that $\mathbb{K}$ is a positive commutative monoid and let $R(X)$ and $S(Y)$ be two consistent $\mathbb{K}$-relations. We say that a $\mathbb{K}$-relation $W(XY)$ is *a largest consistency witness for $R(X)$ and $S(Y)$* if for every consistency witness $U(XY)$ for $R(X)$ and $S(Y)$, we have $U' \subseteq W'$, where U', W' are the supports of $U(XY)$, $W(XY)$. In words, a largest consistency witness for two $\mathbb{K}$-relations is a consistency witness of largest support.

For arbitrary positive commutative monoids, largest consistency witnesses need not exist. A case in point is the bag monoid $\mathbb{N} = (N, +, 0)$. Specifically, consider the two bags $R(X) = \{a{:}1, b{:}1\}$ and $S(Y) = \{c{:}1, d{:}1\}$. These two bags are consistent, but their only two consistency witnesses are $W_1(XY) = \{ac{:}1, bd{:}1\}$ and $W_2(XY) = \{ad{:}1, bc{:}1\}$, which have incomparable supports. Consider also the positive commutative monoid $\mathbb{N}_2 = (\{0, 1, 2\}, \oplus, 0)$, where $1 \oplus 1 = 1 \oplus 2 = 2 \oplus 1 = 2 \oplus 2 = 2$, and 0 is the neutral element of $\oplus$. The same bags as above, but now viewed as $\mathbb{N}_2$-relations are an example of two consistent $\mathbb{N}_2$-relations with no largest consistency witness. Note that the monoid $\mathbb{N}_2$ is finite, while the monoid $\mathbb{N}$ is infinite.

Nonetheless, the property of standard consistent relations having largest consistency witnesses generalizes to relations over idempotent monoids, where a monoid $\mathbb{K} = (K, +, 0)$ is *idempotent* if the identity $x + x = x$ holds, for every $x \in K$.

PROPOSITION 3. *Let $\mathbb{K}$ be an idempotent and positive commutative monoid. Then, for every two consistent $\mathbb{K}$-relations, there is a largest consistency witness.*

Let $\mathbb{K}$ be such a monoid. If $R(X)$ and $S(Y)$ are two consistent $\mathbb{K}$-relations with consistency witnesses $W_1(XY)$ and $W_2(XY)$, then the $\mathbb{K}$-relation $T(XY)$ defined by $T(t) = W_1(t) + W_2(t)$ for every XY-tuple t is also a consistency witness for $R(X)$ and $S(Y)$. Indeed, for every X-tuple r and every Y-tuple s, we have

$$\begin{aligned} T(r) &= W_1(r) + W_2(r) = R(r) + R(r) = R(r) \\ T(s) &= W_1(s) + W_2(s) = S(s) + S(s) = S(s). \end{aligned}$$

Therefore, $T[X] = R$ and $T[Y] = S$, so T is a consistency witness for R and S. We now claim that since $\mathbb{K}$ is positive and since $\mathbb{K}$-relations have (by definition) finite support, there is a consistency witness $W(XY)$ of largest support, which is then a largest consistency witness for $R(X)$ and $S(Y)$.

To see why this claim is true, suppose that $\mathbb{K}$ is positive and idempotent and that $R(X)$ and $S(Y)$ are consistent $\mathbb{K}$-relations. Let N be the number of tuples in the standard join of the standard relations $R' \bowtie S'$, where R' and S' are the supports of R and S. Since, by positivity, every consistency witness for R and S has its support in $R' \bowtie S'$, there are at most 2^N possible supports of witnesses of consistency for R and S. Let $M \leq 2^N$ be the number of such different supports and let $W_1, W_2, \ldots, W_M$ be a collection of witnesses of consistency such that their list of supports $W_1', W_2', \ldots, W_M'$ is the complete enumeration of all supports of witnesses of consistency. Now, consider the $\mathbb{K}$-relation $W(XY)$ defined by $W(t) = \sum_{i=1}^{M} W_i(t)$ for every XY-tuple t, where the sum is in $\mathbb{K}$. By the idempotency of $\mathbb{K}$, the $\mathbb{K}$-relation W is a consistency witness of R and S. And by the positivity of $\mathbb{K}$, the support W' of W contains the support W_i' of every W_i, and hence the support of any consistency witness for $R(X)$ and $S(Y)$ by the choice of the enumeration $W_1, W_2, \ldots, W_M$.

In addition to the Boolean monoid $\mathbb{B} = (\{0,1\}, \vee, 0)$, examples of idempotent monoids include the monoids $\mathbb{T} = (R \cup \{\infty\}, \min, \infty)$, $\mathbb{V} = ([0,1], \max, 0)$, and $\mathbb{P}(A) = (\mathcal{P}(A), \cup, \emptyset)$ introduced in Section 2.

Note that, unlike the case of standard relations, largest consistency witnesses need not be unique for relations over arbitrary idempotent monoids. To see this, consider the positive commutative monoid $\mathbb{L} = (Q^{\geq 0}, \max, 0)$ of non-negative rationals with maximum as operation, and 0 as neutral element, which is idempotent. Consider also the $\mathbb{L}$-relations $R(X) = \{a{:}1, b{:}1\}$ and $S(Y) = \{c{:}1, d{:}1\}$ with disjoint sets of attributes. These two $\mathbb{L}$-relations are consistent and have largest consistency witnesses, namely, any $\mathbb{L}$-relation $W(XY)$ of the form $\{ac{:}1, ad{:}p, bc{:}p, bd{:}1\}$ with $p \in (0,1]$ is a consistency witness for R and S. Thus, while the largest witnesses W are "canonical" in terms of *support*, they need not be "canonical" when taking the *annotations* into account.

There is another sense, however, in which idempotent positive commutative monoids admit canonical-looking consistency witnesses, in addition to having largest support. As discussed earlier, for every positive monoid $\mathbb{K} = (K, +, 0)$, there is a partial preorder $\leq$ on K defined by declaring that $x \leq y$ holds if and only if there exists $z \in K$ such that $x + z = y$. What holds for idempotent positive commutative monoids is that for every two consistent $\mathbb{K}$-relations $R(X)$ and $S(Y)$ and for every *finite* collection of consistency witnesses $U_1, \ldots, U_n$ there is a consistency witness $W(XY)$, still of largest support among all witnesses of consistency of R and S, such that $U_j(t) \leq W(t)$ holds in the preorder $\leq$ of $\mathbb{K}$, for every $j = 1, \ldots, n$ and every XY-tuple t. For this, simply ensure that all target witnesses $U_1, \ldots, U_n$ appear in the enumeration $W_1, W_2, \ldots, W_M$ featuring in the construction of W of the previous paragraph, perhaps by taking M to be an additive term n larger than it was, if necessary. Since by positivity the inequality $U_j(t) \leq \sum_{i=1}^{M} W_i(t) = W(t)$ holds whenever U_j appears in the enumeration $W_1, W_2, \ldots, W_M$, the claim follows. It is apparent from this argument that, in this construction, the witness W depends on the finite list $U_1, \ldots, U_n$; however, only the annotations depend on $U_1, \ldots, U_n$ and, therefore, W is still largest (with respect to support). As regards annotations, the set of consistency witnesses is *dense*: for every finite collection of consistency witnesses $U_1, \ldots, U_n$, there is a largest consistency witness W that sits simultaneously *above* all of them, point-wise in the preorder $\leq$, i.e., W satisfies $U_i(t) \leq W(t)$ for all $i = 1, \ldots, n$ and all XY-tuples t.

Furthermore, there is a case of special interest where not even the annotations of W need depend on the finite list $U_1, \ldots, U_n$. Specifically, if the monoid $\mathbb{K}$ is *finite*, then not only there is a finite number $M \leq 2^N$ of supports of consistency witnesses, but there is just a finite number of consistency witnesses overall. Thus, if in the construction of W we take M to be the total number of consistency witnesses and we let $W_1, W_2, \ldots, W_M$ to be the complete enumeration of these witnesses, then the resulting W is uniquely determined and sits simultaneously above *all* consistency witnesses.

Finally, we note that idempotency is not a necessary condition for the existence of largest consistency witnesses. Indeed, let $\mathbb{R}^{\geq 0} = (R^{\geq 0}, +, 0)$ be the monoid of the non-negative real numbers with addition. This monoid is the reduct of a semifield, namely, the semifield of non-negative reals with the standard addition and multiplication of real numbers, and the standard division by non-zero real numbers as inverse for the multiplication. The reduct $\mathbb{R}^{\geq 0}$ is a positive commutative monoid that has the transportation property but is not, of course, idempotent. Now consider two consistent $\mathbb{R}^{\geq 0}$-relations $R(X)$ and $S(Y)$. As with every other positive commutative monoid that arises from a semifield, their Vorob'ev join $W = R \bowtie_{\mathrm{V}} S$ as defined in Equation (6) is a consistency witness for R and S. It is easy to check that if $U(XY)$ is some other consistency witness, then $W(t) = 0$ implies $U(t) = 0$: indeed, by the absence of zero-divisors in any semifield, the multiplication in (6) gives $W(t) = 0$ only if $R(t[X]) = 0$ or $S(t[Y]) = 0$. Thus, by combining the positivity of the monoid with the fact that $U[X] = R$ and $U[Y] = S$, we get that $W(t) = 0$ only if $U(t) = 0$. This shows that $U' \subseteq W'$

and hence W has largest support. Indeed, in this case W is even "canonical" in its annotations because they depend only on R and S.

An open problem arising from the preceding discussion is to characterize the positive commutative monoids for which largest consistency witnesses always exist.

Acknowledgments Part of the research on this article was carried out at the Simons Institute for the Theory of Computing, where the authors were visiting in the spring and the fall of 2023. Atserias was partially supported by grant no. PID2022-138506NB-C22 (PROOFS BEYOND) and the Severo Ochoa and María de Maeztu Program for Centers and Units of Excellence in R&D (CEX2020-001084-M) of the AEI, and the CERCA and ICREA Academia Programmes of the Generalitat.

9 References

[1] A. Atserias and P. G. Kolaitis. Structure and complexity of bag consistency. In L. Libkin, R. Pichler, and P. Guagliardo, editors, *PODS'21: Proceedings of the 40th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Virtual Event, China, June 20-25, 2021*, pages 247–259. ACM, 2021.

[2] A. Atserias and P. G. Kolaitis. Consistency of relations over monoids. *Journal of the ACM*, 72(3, Article 18):47 pages, 2025. Earlier version in Proc. ACM Manag. Data, 2(2):107, 2024.

[3] C. Beeri, R. Fagin, D. Maier, and M. Yannakakis. On the desirability of acyclic database schemes. *J. ACM*, 30(3):479–513, July 1983.

[4] K. M. Dannert, E. Grädel, M. Naaf, and V. Tannen. Semiring provenance for fixed-point logic. In C. Baier and J. Goubault-Larrecq, editors, *29th EACSL Annual Conference on Computer Science Logic, CSL 2021, January 25-28, 2021, Ljubljana, Slovenia (Virtual Conference)*, volume 183 of *LIPIcs*, pages 17:1–17:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[5] R. Fagin. Degrees of acyclicity for hypergraphs and relational database schemes. *J. ACM*, 30(3):514–550, 1983.

[6] E. Grädel and V. Tannen. Semiring provenance for first-order model checking. *CoRR*, abs/1712.01980, 2017.

[7] T. J. Green. Containment of conjunctive queries on annotated relations. *Theory Comput. Syst.*, 49(2):429–459, 2011.

[8] T. J. Green, G. Karvounarakis, and V. Tannen. Provenance semirings. In L. Libkin, editor, *Proceedings of the Twenty-Sixth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 11-13, 2007, Beijing, China*, pages 31–40. ACM, 2007.

[9] P. Honeyman, R. E. Ladner, and M. Yannakakis. Testing the universal instance assumption. *Inf. Process. Lett.*, 10(1):14–19, 1980.

[10] G. Karvounarakis and T. J. Green. Semiring-annotated data: queries and provenance? *SIGMOD Rec.*, 41(3):5–14, 2012.

[11] M. A. Khamis, H. Q. Ngo, R. Pichler, D. Suciu, and Y. R. Wang. Convergence of datalog over (pre-) semirings. *J. ACM*, 71(2):8:1–8:55, 2024.

[12] E. V. Kostylev, J. L. Reutter, and A. Z. Salamon. Classification of annotation semirings over containment of conjunctive queries. *ACM Trans. Database Syst.*, 39(1):1:1–1:39, 2014.

[13] R. Ramakrishnan. *Database Management Systems*. WCB/McGraw-Hill, 1998.