Database Constraints and Homomorphism Dualities*

Balder ten Cate[1], Phokion G. Kolaitis[1,2], and Wang-Chiew Tan[2,1]

[1] University of California Santa Cruz
[2] IBM Research-Almaden

Abstract. Global-as-view (GAV) constraints form a class of database constraints that has been widely used in the study of data exchange and data integration. Specifically, relationships between different database schemas are commonly described by a schema mapping consisting of a finite set of GAV constraints. Such schema mappings can be viewed as representations of an infinite set of data examples. We study the following problem: when is finite set of GAV constraints uniquely characterizable via a finite set of data examples? By establishing a tight connection between this problem and homomorphism dualities, we obtain a simple criterion for unique characterizability. We also pinpoint the computational complexity of the corresponding decision problem.

1 Introduction and Summary of Results

Since the early days of the relational data model, constraints have played a major role in both the theory and the practice of database systems. In the 1970s and the 1980s, several different types of database constraints, also known as *database dependencies*, were introduced and studied; these include functional dependencies, inclusion dependencies, multi-valued dependencies, and several other classes of dependencies that were used to capture a variety of semantic restrictions that the allowable data must satisfy (see [1] for a survey). In recent years, database dependencies have been used to formalize and study different facets of information integration, which is the problem of accessing and processing data residing in multiple heterogeneous sources. Two prominent facets of information integration are data exchange and data integration (see the surveys [2] and [3]). A key role in the formalization of both data exchange and data integration, as well as of other information integration tasks, is played by the notion of a *schema mapping*. Intuitively, a schema mapping is a specification that describes the relationships between two database schemas, a *source schema* and a *target schema*. More precisely, a schema mapping is a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ with $\mathbf{S}$ a source schema, $\mathbf{T}$ a target schema disjoint from $\mathbf{S}$, and Σ a finite set of constraints involving the schemas $\mathbf{S}$ and $\mathbf{T}$. The constraints in Σ are typically expressed as formulas of a logical formalism. In particular, the class of *source-to-target tuple-generating dependencies* (in short, *s-t tgds*) is the most extensively studied and widely used collection of schema mapping constraints to date, as it strikes a good balance between expressive power and desirable algorithmic properties. By definition, an s-t tgd is a first-order formula of the form

$$\forall \mathbf{x}(\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})),$$

* Ten Cate, Kolaitis, and Tan are supported by NSF grant IIS-0430994 and NSF grant IIS-0905276. Tan is also supported by a NSF CAREER award IIS-0347065.

where $\varphi(\mathbf{x})$ is a conjunction of atoms over $\mathbf{S}$, each variable in $\mathbf{x}$ occurs in at least one atom in $\varphi(\mathbf{x})$, and $\psi(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms over $\mathbf{T}$ with variables in $\mathbf{x}$ and $\mathbf{y}$ (here, an *atom* is a formula $P(x_1, \ldots, x_m)$, where P is a relation symbol and $x_1, \ldots, x_m$ are variables, not necessarily distinct). Intuitively, an s-t tgd asserts that whenever a certain "pattern" is realized in the source, then another "pattern" must be realized in the target. Schema mappings specified by s-t tgds contain as important special cases the class of *global-as-view* (GAV) schema mappings and the class of *local-as-view* (LAV) schema mappings; both GAV and LAV schema mappings are widely used and are supported by many information integration tools. A GAV schema mapping is a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ such that every constraint in Σ is an s-t tgd in which the right-hand side consists of a single atom, that is, it has the form

$$\forall \mathbf{x}(\varphi(\mathbf{x}) \rightarrow T(\mathbf{x})),$$

where $T(\mathbf{x})$ is an atom over the target schema. Intuitively, a GAV constraint specifies that a target relation is described in terms of certain source relations. A LAV schema mapping is a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ such that every constraint in Σ is an s-t tgd in which the left-hand side consists of a single atom, that is, it has the form

$$\forall \mathbf{x}(S(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})),$$

where $S(\mathbf{x})$ is an atom over the source schema. Intuitively, a LAV constraint specifies that a source relation is described in terms of certain target relations. For example, suppose that we wish to form a target relation by deleting the last column from a ternary source relation. This is captured by an s-t tgd of the form $\forall x, y, z(S(x, y, z) \rightarrow U(x, y))$; note that this is both a GAV and a LAV constraint. Similarly, suppose that we wish to form a target relation by appending a column to some binary source relation. This is captured by an s-t tgd of the form $\forall x, y(R(x, y) \rightarrow \exists z T(x, y, z))$, which is a LAV constraint, but not a GAV constraint. Finally, suppose that we wish to form a target relation by joining two binary source relations along the second column of the first relation and the first column of the second. This is captured by an s-t tgd of the form $\forall x, y, z(P(x, y) \wedge R(y, z) \rightarrow T(x, y, z))$, which is a GAV constraint, but not a LAV constraint.

Background on Schema Mappings and Data Examples Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping in which Σ is a finite set of s-t tgds. A *data example* is a pair (I, J) such that I is a source database and J is a target database. If a data example (I, J) satisfies every s-t tgd in Σ, then we say that J is a *solution for I w.r.t. $\mathcal{M}$*. Every schema mapping $\mathcal{M}$ gives rise to the following *data exchange* problem: given a source database I, construct a solution J for I w.r.t. $\mathcal{M}$. In general, a source database I may have an infinite number of solutions. This raises the question: which solution J for I should we chose to materialize in solving the data exchange problem? This question was addressed in [4], where the notion of a *universal solution* was introduced. By definition, a universal solution J for I w.r.t. a schema mapping $\mathcal{M}$ is a solution J for I such that for every solution K for I w.r.t. $\mathcal{M}$, there is a (not necessarily surjective) homomorphism $h : J \rightarrow K$ that is constant on every element of J occurring in I. Intuitively, a universal solution for I is a "most general" solution for I; moreover, a universal solution represents, in a precise technical sense, the entire space of solutions for I. Finally, as shown in [4], given a source database I, a *canonical* universal solution for I can be constructed in time bounded by a polynomial in the size of I using the *chase procedure*. By now, universal solutions have become the standard semantics in data exchange (see [5] for a recent survey).

A schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ is a finite set of s-t tgds, is a syntactic object that provides a finite representation for the infinite space

$$\{(I, J) : (I, J) \text{ is a data example and } J \text{ is a solution for } I \text{ w.r.t. } \mathcal{M}\}.$$

In [6], the following problem was investigated: can this infinite space of data examples be "captured" by a finite set of data examples? The motivation for this problem is that, since schema mappings arising in real-life applications can be quite complex, one would like to use "good" data examples that illustrate the schema mapping at hand and aid in its understanding and refinement. The problem of "capturing" a schema mapping by finitely many data examples was formalized by introducing the notion of *unique characterizability* of a schema mapping via a finite set of data examples of a certain type w.r.t. to a class of s-t tgds. The main focus of [6] was on *universal examples*, where a universal example for $\mathcal{M}$ is a data example (I, J) such that J is a universal solution for I w.r.t. $\mathcal{M}$. In this case, the concept of unique characterizability takes the following precise form. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping, let $\mathcal{C}$ be a class of s-t tgds such that $\Sigma \subseteq \mathcal{C}$, and let $\mathcal{U}$ be a finite set of universal examples for $\mathcal{M}$. We say that $\mathcal{M}$ is *uniquely characterized by $\mathcal{U}$ w.r.t. to $\mathcal{C}$* if, whenever $\mathcal{M}' = (\mathbf{S}, \mathbf{T}, \Sigma')$ is a schema mapping such that $\Sigma' \subseteq \mathcal{C}$ and every data example in $\mathcal{U}$ is a universal example for $\mathcal{M}'$, then Σ' is logically equivalent to Σ. In other words, up to logical equivalence, $\mathcal{M}$ is the only schema mapping with s-t tgds from $\mathcal{C}$ for which $\mathcal{U}$ is a set of universal examples. One of the main results in [6] is that every LAV schema mapping is uniquely characterized by a finite set of universal examples w.r.t. the class of all LAV constraints. On the other hand, it is also shown in [6] that there are natural GAV schema mappings that cannot be uniquely characterized by any finite set of universal examples w.r.t. to the class of GAV schema mappings. It should be noted that the proof of this result made use of a generalization ([7, Theorem 3.15]) of Erdős' well known theorem asserting the existence of graphs of arbitrary large girth and chromatic number.

Summary of Results Our aim in this paper is to address the following questions. Which GAV schema mappings are uniquely characterizable by a finite set of universal examples w.r.t. the class of all GAV schema mappings? Is there an algorithm to tell whether or not a given GAV schema mapping is uniquely characterizable by a finite set of universal examples w.r.t. the class of all GAV schema mappings? If so, what is the exact complexity of this problem? For simplicity, from now on, the term "uniquely characterizable" will mean uniquely characterizable by a finite set of universal examples w.r.t. the class of all GAV schema mappings. Our first main result yields a necessary and sufficient condition for a GAV schema mapping to be uniquely characterizable. This criterion unveils a tight (and rather unexpected) connection between unique characterizability and homomorphism dualities, which we describe in what follows.

Informally, a homomorphism duality is an equivalence between the existence of a homomorphism *to* a structure and the non-existence of a homomorphism *from* the same

Source schema: {Manages}; Target schema: {CEO, TopManager}

GAV constraints:

(σ_1) $\quad \forall x\,(\text{Manages}(x,x) \rightarrow \text{CEO}(x))$

(σ_2) $\quad \forall x,y,z\,(\text{Manages}(x,x) \wedge \text{Manages}(x,y) \wedge \text{Manages}(y,z) \rightarrow \text{TopManager}(y))$

Fig. 1. Example of a GAV schema mapping

structure. The prototypical example of a homomorphism duality is the well-known characterization of 2-colorability: for every graph G, there is a homomorphism from some odd cycle C_{2k+1} *to* G if and only if there is no homomorphism *from* G to the complete graph K_2 with two nodes. Let $\rightarrow$ be the existence-of-a-homomorphism relation between structures over the same schema, i.e., $B \rightarrow C$ means that there is a homomorphism from B to C. Assume that $\mathcal{F}$ and $\mathcal{D}$ are two collections of structures over the same schema. Following [8], we say that the pair $(\mathcal{F};\mathcal{D})$ is a *homomorphism duality* if for every structure A, there exists a structure $F \in \mathcal{F}$ such that $F \rightarrow A$ if and only if there is no structure $D \in \mathcal{D}$ such that $A \rightarrow D$; in symbols, $\bigcup_{F\in\mathcal{F}}(F{\rightarrow}) = \bigcap_{D\in\mathcal{D}}({\not\rightarrow}D)$. If $(\mathcal{F};\mathcal{D})$ is a homomorphism duality, then we say that $\mathcal{F}$ is an *obstruction set for D*. Thus, the aforementioned characterization of 2-colorability is equivalent to the assertion that the pair $(\{C_{2k+1} : k \geq 1\}; \{K_2\})$ is a homomorphism duality; moreover, $\{C_{2k+1} : k \geq 1\}$ is an obstruction set for $\{K_2\}$.

For every GAV schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and every relation symbol T in $\mathbf{T}$, we construct a finite set $\mathcal{F}_{\mathcal{M},T}$ of relational structures over a signature consisting of the relations from the source schema $\mathbf{S}$ plus finitely many constant symbols, and show that $\mathcal{M}$ is uniquely characterizable if and only if each $\mathcal{F}_{\mathcal{M},T}$ is an obstruction set for some finite set $\mathcal{D}_{\mathcal{M},T}$ of structures. This result provides the aforementioned necessary and sufficient condition for unique characterizability of GAV schema mappings; however, it does not yield immediately an algorithm for testing whether or not a given schema mapping is uniquely characterizable. In [8], it was shown that a finite set $\mathcal{F}$ of homomorphically incomparable core structures is an obstruction set for some finite set $\mathcal{D}$ if and only if every structure in $\mathcal{F}$ obeys a certain *acyclicity* condition. This result holds for structures over a signature with relation symbols but no constant symbols. Here, we show that this characterization can be extended to structures over a signature with both relation symbols and constant symbols. In particular, we show that a GAV schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is uniquely characterizable if and only if every structure in the aforementioned finite sets $\mathcal{F}_{\mathcal{M},T}$ obeys a weaker acyclicity condition, which we call *c-acyclicity*. Moreover, there is an algorithm that, given such a GAV schema mapping, computes a uniquely characterizing set of examples. Informally, the c-acyclicity condition allows for cycles, but every cycle must contain (the interpretation of) a constant symbol. This gives rise to a powerful tool for determining whether or not a given GAV schema mapping is uniquely characterizable. As an illustration of the power of this tool, it follows immediately that the GAV schema mapping $\mathcal{M}$ specified by the single s-t tgd $\forall x,y,z(E(x,z) \wedge E(z,y) \rightarrow P(x,y))$ is uniquely characterizable. In contrast, the GAV schema mapping $\mathcal{M}'$ specified by the single s-t tgd $\forall x,y,z,w(E(x,z) \wedge E(z,y) \wedge E(w,w) \rightarrow P(x,y))$ is not uniquely characterizable.

Finally, from a computational-complexity standpoint, we show that the following problem is NP-complete: given a GAV schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, is it uniquely characterizable? We also show that the computational complexity of this problem drops down to LOGSPACE, if $\mathcal{M}$ is in a certain normalized form in which the left-hand sides of the s-t tgds in Σ are cores. In addition, we obtain results concerning the decidability and computational complexity of several other natural algorithmic problems involving GAV schema mappings, universal examples, and unique characterizability.

Most proofs in this paper are omitted for lack of space.

2 Basic Concepts and Preliminaries

Signatures, Structures, Schemas and Databases In logic, a *signature* is a collection of relation symbols, function symbols, and constant symbols. Here, we will be concerned only with signatures consisting of finitely many relation symbols $R_1, \ldots, R_n$ of designated arities and finitely many constant symbols $c_1, \ldots, c_k$. A *structure* A over such a signature is a tuple $A = (D, R_1^A, \ldots, R_n^A, c_1^A, \ldots, c_k^A)$, where D is a set, called the *domain* of A, each R_i^A is a relation on D whose arity matches the arity of the relation symbol R_i, and each c_j^A is an element of D. If no constant symbols are present, then we talk about a *relational signature* and about *relational structures* over that signature. In what follows, we will assume that all structures considered are finite, that is, the domain and the relations of the structure are finite. For simplicity of notation and when the structure A at hand is understood from the context, we will often use R_i to denote both the relation symbol R_i and the relation R_i^A interpreting it on A.

In databases, a *schema* is a finite collection of relation symbols $R_1, \ldots, R_n$ of designated arities, i.e., a schema is a relational signature. A *database* I over such a schema is a tuple $I = (R_1^I, \ldots, R_n^I)$ of finite relations over some domain. Every database I can be identified with a relational structure $(\mathrm{adom}(I), R_1^I, \ldots, R_n^I)$, where $\mathrm{adom}(I)$ is the *active domain* of I, that is, the set of all values occurring in the relations of I. In what follows, we will use relational structures and databases in an interchangeable way.

A *homomorphism* from A to B is a function h from the domain of A to the domain of B such that for every relation symbol R_i and every constant symbol c_j: (1) if $(a_1, \ldots, a_m) \in R_i^A$, then $(h(a_1), \ldots, h(a_m)) \in R_i^B$; and (2) $h(c_j^A) = c_j^B$.

Schema Mappings and Universal Solutions A *schema mapping* is a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ and $\mathbf{T}$ are disjoint schemas, called the *source* schema and the *target* schema, and Σ is a set of source-to-target tuple generating dependencies, as defined in Section 1. If Σ consists entirely of GAV constraints, then $\mathcal{M}$ is called a GAV schema mapping; if Σ consists entirely of LAV constraints, then $\mathcal{M}$ is a LAV schema mapping. In this paper, our main focus will be on GAV schema mappings.

Figure 1 contains an example of a GAV schema mapping; it will be our running example throughout paper. In this example, the source database contains information about managerial relationships in a company using a binary relation "Manages", while the target database contains information about managerial roles, using the unary relations "CEO" and "TopManager". Incidentally, note that σ_1 is both a GAV constraint and a LAV constraint, while σ_2 is a GAV constraint but not a LAV constraint.

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping. A *data example* is a pair (I, J), where I is a source database (i.e., a database over $\mathbf{S}$) and J is a target structure (i.e., a database over $\mathbf{T}$). If I is a source database, then a *solution for I w.r.t. $\mathcal{M}$* is a target database J such that the data example (I, J) satisfies every constraint in Σ. A *universal solution* *for I w.r.t.* $\mathcal{M}$ is a solution J for I w.r.t. $\mathcal{M}$ such that for every solution J' for I w.r.t $\mathcal{M}$, there is a homomorphism $h : J \to J'$ such that h is constant on the active domain $\mathrm{adom}(I)$ of I, i.e., $h(a) = a$, for each $a \in \mathrm{adom}(I) \cap \mathit{dom}(h)$. It was shown in [4] that if $\mathcal{M}$ is any schema mapping specified by s-t tgds, then every source structure I has a *canonical* universal solution $\text{CanSol}_{\mathcal{M}}(I)$, which can be constructed in time bounded by a polynomial in the size of I. If $\mathcal{M}$ is a GAV schema mapping and I is a source database, then the active domain of $\text{CanSol}_{\mathcal{M}}(I)$ is contained in $\mathrm{adom}(I)$. In fact, in the case of GAV schema mappings, $\text{CanSol}_{\mathcal{M}}(I)$ is the only universal solution for I such that its active domain is contained in $\mathrm{adom}(I)$. Moreover, the relations of $\text{CanSol}(I)$ consist precisely of all tuples that are "dictated" by the GAV constraints of $\mathcal{M}$, in the sense that they are the right-hand-side of a GAV constraint σ of $\mathcal{M}$, under a variable assignment that makes the left-hand-side of σ true in I. Note, however, that the state of affairs is more complicated for non-GAV schema mappings (and, in particular, for LAV schema mappings), since the active domain of $\text{CanSol}(I)$ may contain values not occurring in $\mathrm{adom}(I)$.

Characterizing Schema Mappings via Data Examples In [6], different notions of data examples were considered for "illustrating" the semantics of a schema mapping, such as positive examples, negative examples, and universal examples.

- A *positive example* (I, J) *for a schema mapping* $\mathcal{M}$ is a data example for $\mathcal{M}$ such that J is a solution for I w.r.t. $\mathcal{M}$.
- A *negative example* (I, J) *for a schema mapping* $\mathcal{M}$ is a data example for $\mathcal{M}$ such that J is *not* a solution for I w.r.t. $\mathcal{M}$.
- A *universal example* (I, J) *for a schema mapping* $\mathcal{M}$ is a data example for $\mathcal{M}$ such that J is a universal solution for I w.r.t. $\mathcal{M}$.

Among these, *universal examples* were shown in [6] to be the most promising type of data example for capturing the semantics of a schema mapping. Specifically, the central question studied in [6] is the *unique characterizability* problem: can a schema mapping be "captured" by a finite set of data examples of particular types w.r.t. a class of s-t tgds? In the context of universal examples, the unique characterizability problem is defined as follows. First, if $\mathcal{M}$ is a schema mapping $\mathcal{M}$ and $\mathcal{U}$ is a set of data examples, we say that $\mathcal{M}$ *fits the universal examples* $\mathcal{U}$ if all data examples in $\mathcal{U}$ are universal examples of $\mathcal{M}$. We say that a schema mapping $\mathcal{M}$ is *uniquely characterized by a finite set of universal examples* $\mathcal{U}$ *w.r.t. a class of s-t tgds* $\mathcal{C}$ if $\mathcal{M}$ fits $\mathcal{U}$ and for every finite set $\Sigma' \subseteq \mathcal{C}$ such that $\mathcal{M}' = (\mathbf{S}, \mathbf{T}, \Sigma')$ fits $\mathcal{U}$, we have that Σ and Σ' are logically equivalent. Similar definitions apply in the case of positive examples and negative examples.

The following results were established in [6]. First, there are LAV schema mappings that are not uniquely characterizable by any finite set of positive and negative examples w.r.t. to the class of all LAV constraints. In contrast, every LAV schema mapping is uniquely characterized by some finite set of universal examples w.r.t. the class of all LAV constraints. Moreover, this positive result extends to the much broader classes of n*-modular* schema mappings [9], where n is a positive integer. The state of affairs for GAV schema mappings and universal examples turned out to be quite different, as revealed by the next result.

Theorem 1. ([6]) *The following statements are true.*

- *The schema mapping specified by the GAV constraint* $\forall x, y(S(x, y) \rightarrow T(x, y))$ *is uniquely characterizable w.r.t. the class of GAV constraints by the universal examples given in Figure 2.*
- *The schema mapping specified by the GAV constraint* $\forall x, y, z(S(x, y) \wedge R(z, z) \rightarrow T(x, y))$ *is not uniquely characterizable by any finite set of universal examples w.r.t. the class of GAV constraints.*

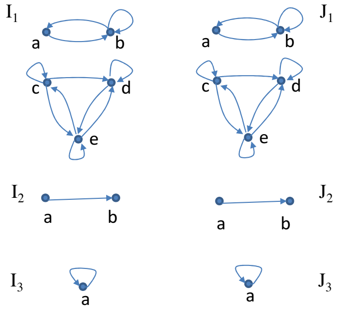


Fig. 2. Universal examples uniquely characterizing the copy constraint $\forall x, y\ (S(x, y) \rightarrow T(x, y))$

The constraint in the first part of Theorem 1 can be thought of as a "copy" constraint that copies the relation S into the relation T. The constraint in the second part of Theorem 1 can be thought of as a "copy constraint with a trigger": S is copied into T, provided the relation R contains a self-loop. What is the reason that these two GAV constraints have such different properties? More generally, which GAV schema mappings are uniquely characterizable via universal examples w.r.t. to the class of all GAV constraints? Is the associated decision problem (whether or not a given GAV schema mapping is uniquely characterizable) decidable?

Before embarking on the study of these questions, we point out that unique characterizability of GAV schema mappings via universal examples w.r.t. the class of all GAV constraints is equivalent to unique characterizability via positive and negative examples.

Proposition 1. *For GAV schema mappings* $\mathcal{M}$*, the following are equivalent w.r.t. the class of all GAV constraints:*

1. $\mathcal{M}$ *is uniquely characterizable by positive and negative examples,*
2. $\mathcal{M}$ *is uniquely characterizable by universal examples,*
3. $\mathcal{M}$ *is uniquely characterizable by positive, negative, and universal examples*

Proposition 1 shows that, in the GAV setting, unique characterizability is a particularly robust notion, in the sense that it does not depend on whether we consider universal examples, or positive and negative examples. As we will focus on GAV schema mappings, we will therefore simply speak of *unique characterizability*, meaning *unique characterizability by a finite set of universal examples w.r.t. the class of GAV constraints*.

3 Homomorphism dualities and unique characterizations

In this section, we establish a connection between unique characterizations of GAV schema mappings on the one hand, and homomorphism dualities for relational structures on the other. Specifically, we show that the problem of testing whether a GAV schema mapping is uniquely characterizable can be reduced to a certain problem concerning the existence of a homomorphism duality; furthermore, the problem of testing whether a GAV schema mapping is uniquely characterized by a given set of universal examples can be reduced to the question of whether a given pair of sets of structures is a homomorphism duality. Since these two problems concerning homomorphism dualities are decidable (cf. Section 4), we will be able to derive decidability results for the two problems concerning unique characterizations (cf. Section 5).

Homomorphism Dualities As described in Section 1, a homomorphism duality is an equivalence between the existence of a homomorphism *to* a structure and the non-existence of a homomorphism *from* the same structure. We will work with a finite signature consisting of relation symbols and constant symbols; recall that all structures considered here are assumed to be finite. Given a structure A, we denote by $A{\to}$ the set of all structures (over the same signature) that A has a homomorphism into; in symbols, $A{\to} = \{B : A \to B\}$. Similarly, ${\not\to}A$ is the set of all structures that do not have a homomorphism into A, i.e., ${\not\to}A = \{B : B \not\to A\}$.

Definition 1. *Let $\mathcal{F}$ and $\mathcal{D}$ be two sets of structures. We say that the pair $(\mathcal{F};\mathcal{D})$ is a* homomorphism duality *if $\bigcup_{F\in\mathcal{F}}(F{\to}) = \bigcap_{D\in\mathcal{D}}({\not\to}D)$. If $(\mathcal{F};\mathcal{D})$ is a homomorphism duality, then we say that $\mathcal{F}$ is an* obstruction set *for $\mathcal{D}$.*

If $(\mathcal{F};\mathcal{D})$ is a homomorphism duality, it means that the class of all structures is partitioned into two disjoint subclasses, namely, the subclass $\bigcup_{F\in\mathcal{F}}(F{\to})$ of those structures that some structure in $\mathcal{F}$ has a homomorphism into, and the subclass $\bigcup_{D\in\mathcal{D}}({\to}D)$ of those structures that have a homomorphism into some structure in $\mathcal{D}$. This is illustrated in Figure 3 (where, intuitively, the direction of homomorphisms is upward).

A homomorphism duality in which both sets of structures are singletons is called a *simple homomorphism duality pair*, and is typically written without curly braces. Homomorphism dualities, and in particular simple homomorphism duality pairs, have been studied extensively in graph theory (where they are used to gain understanding of the structure of the lattice of graphs and homomorphisms, cf. [7]) and in the context of constraint satisfaction problems (where they have been used in order to identify classes of tractable constraint satisfaction problems, cf. e.g., [10]).

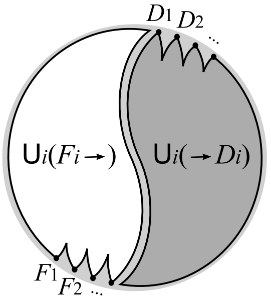


Fig. 3. A homomorphism duality

Homomorphism Dualities and Unique Characterizations We will now establish the fundamental connection between unique characterizations of GAV schema mappings and homomorphism dualities. In order to state the result, we associate a *canonical structure* with every GAV constraint. Specifically, consider a GAV constraint

$$\sigma = \forall x_1, \ldots, x_m(\phi(x_1, \ldots, x_m) \to T(y_1, \ldots, y_k))$$

over a source schema $\mathbf{S} = \{S_1, \ldots, S_n\}$ and a target schema $\mathbf{T} = \{T, \ldots\}$, with $y_1, \ldots, y_k \in \{x_1, \ldots, x_m\}$. The *canonical structure* associated with σ is the following structure A_σ over the signature $\{S_1, \ldots, S_n, c_1, \ldots, c_k\}$:

$$A_\sigma = (\{x_1, \ldots, x_m\}, S_1^{A_\sigma}, \ldots, S_n^{A_\sigma}, c_1^{A_\sigma}, \ldots, c_k^{A_\sigma})$$

where each relation $S_i^{A_\sigma}$ consists of the tuples in the atoms of ϕ that involve S_i, and each $c_j^{A_\sigma} = y_j$. In database-theory terms, A_σ is the canonical instance of the left-hand side of σ (viewed as a conjunctive query), expanded with constant symbols marking the exact sequence of exported variables $y_1, \ldots, y_k$. For a GAV schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and a target relation $T \in \mathbf{T}$, we denote by $\mathcal{F}_{\mathcal{M},T}$ the set of all canonical structures of GAV constraints $\sigma \in \Sigma$ that use the target relation T in their right-hand-side.

Theorem 2. *A GAV schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is uniquely characterizable if and only if for each $T \in \mathbf{T}$, $\mathcal{F}_{\mathcal{M},T}$ is an obstruction set for a finite set of structures.*

Before we present the proof of Theorem 2, let us illustrate the result by revisiting our running example in Figure 1. The canonical structures A_{σ_1} and A_{σ_2} of the GAV constraints σ_1, σ_2 can be depicted as $\boxed{\circlearrowright \cdot \ ^{c_1}}$ and $\boxed{\circlearrowright\cdot \longrightarrow \cdot^{c_1} \longrightarrow \cdot}$, respectively, where an arrow indicates that two elements stand in the Manages relation. Since σ_1 and σ_2 use different target relations, Theorem 2 tells us that, in order to determine whether this GAV schema mapping is uniquely characterizable, it is enough to test whether each of these structures, taken as a singleton set, is an obstruction set for a finite set of structures. As it turns out, $\{A_{\sigma_1}\}$ is indeed an obstruction set for a finite set of structures (in fact, the reader may easily verify that $(A_{\sigma_1}; B)$ is a simple homomorphism duality pair, with B the structure depicted by $\boxed{\cdot^{c_1} \longleftrightarrow \cdot\circlearrowleft}$). On the other hand, $\{A_{\sigma_2}\}$ is not an obstruction set for any finite set of structures, as will follow from results presented in Section 4. It follows that our example schema mapping is not uniquely characterizable.

We will now proceed with the proof of Theorem 2. We will use the following convenient notation, familiar from logic. If A is a structure over the signature $\{S_1, \ldots, S_n\}$, and $a_1, \ldots, a_k$ is a sequence of (not necessarily distinct) elements of the domain of A, then we denote by $\langle A, a_1, \ldots, a_k \rangle$ the structure over the signature $\{S_1, \ldots, S_n, c_1, \ldots c_k\}$ that has the same domain as A and agrees with A on the denotation of the relations $S_1, \ldots, S_n$, and in which each constant symbol c_i denotes the element a_i. In other words, $\langle A, a_1, \ldots, a_k \rangle$ is identical to A except that the elements $a_1, \ldots, a_k$ are named using fresh constant symbols.

Proof (of Theorem 2). First, we show that, when it comes to the question of unique characterizability, we can restrict attention to schema mappings for a single target relation. This is stated by the next lemma. For any relation $T \in \mathbf{T}$, we denote by $\mathcal{M}|_T$ the schema mapping $(\mathbf{S}, \{T\}, \Sigma')$ where $\Sigma' \subseteq \Sigma$ consists of all GAV constraints whose right-hand side contains the target relation T.

Lemma 1. *$\mathcal{M}$ is uniquely characterizable if and only if for each $T \in \mathbf{T}$, the schema mapping $\mathcal{M}|_T$ is uniquely characterizable.*

We will also make use of the following fact concerning canonical universal solutions of GAV schema mappings (cf. [9]):

Lemma 2. *For all source structures I_1, I_2, every homomorphisms $h : I_1 \to I_2$ is also a homomorphism $h : \mathrm{CanSol}_{\mathcal{M}}(I_1) \to \mathrm{CanSol}_{\mathcal{M}}(I_2)$.*

We now proceed with the main proof. By Lemma 1, we may assume $\mathbf{T} = \{T\}$, and show that $\mathcal{M}$ is uniquely characterizable if and only if $\mathcal{F}_{\mathcal{M},T}$ is an obstruction set for a finite set of structures.

($\Rightarrow$) Let $\mathcal{U}$ be a set of universal examples uniquely characterizing $\mathcal{M}$. Let $\mathcal{D}$ be the set $\{\langle I, \mathbf{a} \rangle \mid (I, J) \in \mathcal{U}, \mathbf{a} \in dom(I)^k \setminus T^J\}$. We claim that $(\mathcal{F}_{\mathcal{M},T}; \mathcal{D})$ is a homomorphism duality. To see this, it is enough to observe that, for all source structures I and for all tuples $\mathbf{a}$, we have that:

- $\langle F, \mathbf{b}\rangle \to \langle I, \mathbf{a}\rangle$ for some $\langle F, \mathbf{b}\rangle \in \mathcal{F}_{\mathcal{M},T}$ if and only if $T(\mathbf{a}) \in \text{CanSol}_{\mathcal{M}}(I)$,
- $\langle I, \mathbf{a}\rangle \to \langle D, \mathbf{b}\rangle$ for some $\langle D, \mathbf{b}\rangle \in \mathcal{D}$ if and only if $T(\mathbf{a}) \notin \text{CanSol}_{\mathcal{M}}(I)$.

The first item follows immediately from the construction of $\mathcal{F}$. The left-to-right direction of the second item follows from Lemma 2. The right-to-left direction of the second item can be shown by contradiction: suppose $\text{CanSol}_{\mathcal{M}}(I)$ does not contain $T(\mathbf{a})$ and $\langle I, \mathbf{a}\rangle$ does not homomorphically map into any $\langle D, \mathbf{b}\rangle \in \mathcal{D}$. Let $\mathcal{M}'$ extend $\mathcal{M}$ with an extra GAV constraint, namely the canonical GAV constraint of $\langle I, \mathbf{a}\rangle$. Clearly, $\mathcal{M}'$ is not logically equivalent to $\mathcal{M}$, but it is not hard to see that $\mathcal{M}'$ fits the universal examples $\mathcal{U}$, contradicting the fact that the universal examples $\mathcal{U}$ uniquely characterize $\mathcal{M}$.

($\Leftarrow$) Let $\mathcal{D}$ be a finite set of structures such that $(\mathcal{F}_{\mathcal{M},T}; \mathcal{D})$ is a homomorphism duality. Let $\mathcal{U} = \{(I, \text{CanSol}_{\mathcal{M}}(I)) \mid \langle I, \mathbf{a}\rangle \in \mathcal{F}_{\mathcal{M},T} \cup \mathcal{D}\}$. We claim that $\mathcal{U}$ uniquely characterizes $\mathcal{M}$. For, consider any schema mapping $\mathcal{M}'$ fitting the universal examples in $\mathcal{U}$, any source structure I and any k-tuple $\mathbf{a}$ of elements from the domain of I, where k is the arity of T. There are two cases:

The first case is where $\langle F, \mathbf{b}\rangle \to \langle I, \mathbf{a}\rangle$ for some $\langle F, \mathbf{b}\rangle \in \mathcal{F}_{\mathcal{M},T}$. By construction of $\mathcal{F}_{\mathcal{M},T}$, we have that $\text{CanSol}_{\mathcal{M}}(F)$, hence also $\text{CanSol}_{\mathcal{M}'}(F)$, contains $T(\mathbf{b})$. It follows by Lemma 2 that $\text{CanSol}_{\mathcal{M}}(I)$ and $\text{CanSol}_{\mathcal{M}'}(I)$ contain $T(\mathbf{a})$.

The second case is where $\langle I, \mathbf{a}\rangle \to \langle D, \mathbf{b}\rangle$ for some $\langle D, \mathbf{b}\rangle \in \mathcal{D}$. It follows from the duality and from the construction of $\mathcal{F}_{\mathcal{M},T}$ that $\text{CanSol}_{\mathcal{M}}(D)$, and therefore also $\text{CanSol}_{\mathcal{M}'}(D)$, does not contain $T(\mathbf{b})$. It follows by Lemma 2 that $\text{CanSol}_{\mathcal{M}}(I)$ and $\text{CanSol}_{\mathcal{M}'}(I)$ both do not contain $T(\mathbf{a})$.

This shows that $\text{CanSol}_{\mathcal{M}}(I) = \text{CanSol}_{\mathcal{M}'}(I)$. In other words, $\mathcal{M}$ and $\mathcal{M}'$ are logically equivalent, and hence $\mathcal{M}$ is uniquely characterized by $\mathcal{U}$. □

The above result links the notion of unique characterizability to that of being an obstruction set of a finite set of structures. In a similar fashion, we can link unique characterizations themselves to homomorphism dualities. This is expressed by the following Theorem. The proof is a variation of that of Theorem 2.

Theorem 3. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a GAV schema mapping and $\mathcal{U}$ a set of universal examples for $\mathcal{M}$. For each $T \in \mathbf{T}$, let $\mathcal{F}_T = \{\langle I, \boldsymbol{a}\rangle \mid (I, J) \in \mathcal{U},\ \boldsymbol{a} \in T^J\}$ and let $\mathcal{D}_T = \{(\langle I, \boldsymbol{a}\rangle \mid (I, J) \in \mathcal{U},\ \boldsymbol{a} \in dom(I)^k \setminus T^J\}$, where k is the arity of T. Then the following statements are equivalent:*

1. *$\mathcal{U}$ uniquely characterizes $\mathcal{M}$.*
2. *For each $T \in \mathbf{T}$, $(\mathcal{F}_T; \mathcal{D}_T)$ is a homomorphism duality.*

Theorem 2 and 3 reduce questions about unique characterizations to questions about homomorphism dualities. In the remainder of this section, we show that the same applies the other way around (so that we will be able to transfer not only complexity theoretic upper bounds for these questions, but also lower bounds). To state these results, we need a way to associate to each structure a GAV constraint. Given a structure

$$A = (\{a_1, \ldots, a_m\}, S_1^A, \ldots, S_n^A, c_1^A, \ldots, c_k^A)$$

with $c_1^A = a_{i_1}, \ldots, c_k^A = a_{i_k}$, we associate with it the *canonical GAV constraint*

$$\sigma_A = \forall x_1, \ldots, x_m (\bigwedge_{\substack{1 \le i \le n \\ (a_{j_1}, \ldots, a_{j_\ell}) \in S_i^A}} S_i(x_{j_1}, \ldots, x_{j_\ell}) \ \to \ T(x_{i_1}, \ldots, x_{i_k}))$$

over the source schema $\mathbf{S} = \{S_1, \ldots, S_n\}$ and a target schema $\mathbf{T} = \{T\}$, where T is a k-ary target relation. In other words, the antecedent of σ_A is the atomic diagram of (the purely relational part of) A, while the conclusion of σ_A lists the elements denoted by the constant symbols in order. The reader may verify that, by this definition, the canonical GAV constraint of the canonical structure of a GAV constraint σ is just σ itself (up to renaming of variables and reordering of conjuncts). The *canonical GAV schema mapping* $\mathcal{M}_{\mathcal{F}}$ of a finite set of structures $\mathcal{F}$ is the schema mapping defined by the canonical GAV constraints of the structures in $\mathcal{F}$.

Theorem 4. *Let $\mathcal{F}$ be a finite set of structures for a signature $\{S_1, \ldots, S_n, c_1, \ldots, c_k\}$. Then $\mathcal{F}$ is an obstruction set for a finite set of structures if and only if $\mathcal{M}_{\mathcal{F}}$ is uniquely characterizable.*

Theorem 5. *Let $\mathcal{F}, \mathcal{D}$ be finite sets of structures for a signature $\{S_1, \ldots, S_n, c_1, \ldots, c_k\}$. Let $\mathcal{U}_{\mathcal{F},\mathcal{D}}$ be the set of all pairs $(I, \text{CanSol}_{\mathcal{M}_{\mathcal{F}}}(I))$ where $I = (D, S_1^A, \ldots, S_n^A)$ for some $A = (D, S_1^A, \ldots, S_n^A, c_1^A, \ldots, c_k^A) \in \mathcal{F} \cup \mathcal{D}$. The following are equivalent:*

1. *$(\mathcal{F}; \mathcal{D})$ is a homomorphism duality*
2. *$\mathcal{M}_{\mathcal{F}}$ is uniquely characterized by $\mathcal{U}_{\mathcal{F},\mathcal{D}}$, and, moreover, for each structure $A = (D, S_1^A, \ldots, S_n^A, c_1^A, \ldots, c_k^A) \in \mathcal{D}$, $\text{CanSol}_{\mathcal{M}_{\mathcal{F}}}(A)$ does not contain $T(c_1^A, \ldots, c_k^A)$.*

4 Results on Homomorphism Dualities

In this section, we will present a characterization of the finite sets of structures $\mathcal{F}$ that are an obstruction set for a finite set of structures $\mathcal{D}$. For the case of relational signatures without constant symbols, an elegant characterization of such sets $\mathcal{F}$ was established in [8]. Our main contribution in this section is to show that the characterization can be extended in a natural way to structures with constant symbols.

We now introduce some terminology and state two basic facts concerning homomorphism dualities. Recall that, for structures A, B, we write $A \to B$ if there is a homomorphism from A to B. We say that A and B are *homomorphically equivalent* if $A \to B$ and $B \to A$, and we say that A and B are *homomorphically incomparable* if there are neither $A \to B$, nor $B \to A$. Every finite structure A is known to have a unique (up to isomorphism) smallest homomorphically equivalent substructure that is homomorphically equivalent to A, which is known as the *core* of A [11]. A structure is said to be a *core* if it is its own core. For a set X of structures, we denote by $\operatorname{core} X$ the set of cores of structures in X, we denote by $\max X$ any subset $Y \subseteq X$ consisting of homomorphically incomparable structures such that for all $A \in X$, there is a $B \in Y$ with $A \to B$; in a dual manner, we denote by $\min X$ any subset $Y \subseteq X$ consisting of homomorphically incomparable structures such that for all $A \in X$, there is a $B \in Y$ with $B \to A$. If one reflects on the definition of homomorphism dualities, and keeps in mind Figure 3, the following fact becomes evident (note that, if $A \to B$, then $(B \to) \subseteq (A \to)$ and $(\to A) \subseteq (\to B)$):

Fact 6 *Let $\mathcal{F}$ and $\mathcal{D}$ be finite sets of structures. Then $(\mathcal{F}; \mathcal{D})$ is a homomorphism duality if and only if $(\min \operatorname{core} \mathcal{F}; \max \operatorname{core} \mathcal{D})$ is a homomorphism duality.*

By construction, $\min \operatorname{core} \mathcal{F}$ and $\max \operatorname{core} \mathcal{D}$ have the property that they consist of pairwise homomorphically incomparable core structures. Hence, we may restrict attention to sets $\mathcal{F}$ and $\mathcal{D}$ consisting of pairwise homomorphically incomparable core structures.

The second fact states that for any given finite set of structures $\mathcal{F}$, there is at most one finite set of structures $\mathcal{D}$ for which $\mathcal{F}$ is an obstruction set, assuming that $\mathcal{D}$ consists of pairwise incomparable core structures. The proof, which we omit, is elementary, using the definition of homomorphism duality and the fact that homomorphisms compose.

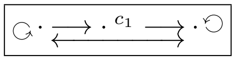
Fact 7 *Let $\mathcal{F}, \mathcal{D}, \mathcal{D}'$ be finite sets of homomorphically incomparable core structures such that $(\mathcal{F}; \mathcal{D})$ and $(\mathcal{F}; \mathcal{D}')$ are homomorphism dualities. Then $\mathcal{D}$ and $\mathcal{D}'$ contain the same structures up to isomorphism.*

Known Results for Structures without Constant Symbols Consider signatures consisting only of relation symbols. The main result from [8] states that a finite set of homomorphically incomparable core structures $\mathcal{F}$ for such a signature is an obstruction set for a finite set of structures $\mathcal{D}$ if and only if each structure from $\mathcal{F}$ obeys a certain acyclicity condition, which we will now define.

A *fact* of a structure A is an expression $R(a_1, \ldots, a_m)$ such that R is one of the relations of A and $(a_1, \ldots, a_m) \in R$. The *incidence graph* $\mathsf{inc}(A)$ of a structure A is the undirected (bi-partite) graph whose vertices are the elements and the facts of A, and with an edge between an element and a fact if the element occurs in the fact. We call a structure A *acyclic* if $\mathsf{inc}(A)$ is acyclic and no fact of A contains the same element twice. Note that the second condition has to be included explicitly in the definition, since it is not implied by the first (however, in [8], an equivalent definition is given in terms of an incidence multi-graph that may contain several edges between the same fact and element, so that the second condition is not needed).

Theorem 8 ([8]). *Consider a signature consisting of relation symbols only, and let $\mathcal{F}$ be a finite set of homomorphically incomparable core structures. Then $\mathcal{F}$ is an obstruction set for a finite set of structures if and only if every structure in $\mathcal{F}$ is acyclic. Moreover, there is an algorithm that, given such a set $\mathcal{F}$ consisting of acyclic structures, computes a finite set $\mathcal{D}$ such that $(\mathcal{F}; \mathcal{D})$ is a homomorphism duality.*

Incidentally, the algorithm for computing $\mathcal{D}$ from $\mathcal{F}$ given in [8] runs in double exponential time, and no matching lower bound is known (cf. also [12] for improved bounds in the special case of simple homomorphism duality pairs). Foniok et al. [8] also consider the problem of testing whether a given finite set of structures $\mathcal{F}$ has a finite obstruction set. They show (for structures without constant symbols) that this problem is NP-complete and that one can effectively compute an obstruction set if it exists.

A Generalization for the Case with Constant Symbols In the presence of constant symbols, acyclicity is no longer a necessary condition for being an obstruction set for a finite set of structures. For instance, consider the structure A depicted by $\boxed{\cdot \longleftrightarrow \cdot}$, and let A' be the expansion of A with a constant symbol c_1 denoting the left-most element, as in $\boxed{{}^{c_1}\cdot \longleftrightarrow \cdot}$. Since the incidence graph of A contains a cycle, by Theorem 8 there is no finite set of structures $\mathcal{D}$ such that $(\{A\}; \mathcal{D})$ is a homomorphism duality. The situation for A' is very different. Indeed, if we let B' be the structure depicted by , then $(A'; B')$ is a simple homomorphism duality pair.

Nevertheless, Theorem 8 can be extended in a natural way to structures with constant symbols. To this end, we call a structure A over a signature consisting of relation symbols and constant symbols *c-acyclic* if the following both hold:

1. Every cycle in $\mathsf{inc}(A)$ passes through an element named by a constant symbol,
2. If a fact of A contains the same element a twice, a is named by a constant symbol.

Note that for structures without constant symbols, c-acyclicity is equivalent to acyclicity. Also note that the structure A' we discussed above is c-acyclic.

The proof of the following Theorem is based on a reduction to Theorem 8.

Theorem 9. *Consider a signature consisting of relation symbols and constant symbols, and let $\mathcal{F}$ be a finite set of homomorphically incomparable core structures. Then $\mathcal{F}$ is an obstruction set for a finite set of structures if and only if every structure in $\mathcal{F}$ is c-acyclic. Moreover, there is an algorithm that, given such a set $\mathcal{F}$ consisting of c-acyclic structures, computes a finite set $\mathcal{D}$ such that $(\mathcal{F}; \mathcal{D})$ is a homomorphism duality.*

From Theorem 9, we can derive the following computability and complexity results (using the fact that undirected reachability is in LOGSPACE [13]).

Corollary 1. *The following problem is* NP*-complete: given a finite set of structures $\mathcal{F}$, determine if $\mathcal{F}$ is an obstruction set for a finite set of structures. The same problem is in* LOGSPACE *if the input is a set of homomorphically incomparable core structures.*

Corollary 2. *The following problem is decidable: given finite sets of structures $\mathcal{F}$ and $\mathcal{D}$, determine if $(\mathcal{F}; \mathcal{D})$ is a homomorphism duality.*

5 An Effective Characterization of Unique Characterizability

We now put the results from the previous sections to use. Our main result is an effective characterization of the uniquely characterizable GAV schema mappings.

We say that a GAV schema mapping $\mathcal{M}$ is *normalized* if (i) the canonical structure of the left-hand-side of each GAV constraint is a core, and (ii) for any two GAV constraints for the same target relation, the canonical structures are homomorphically incomparable. Note that every GAV schema mapping is equivalent to a normalized GAV schema mapping, of the same size or smaller, which can be computed in exponential time (in fact, in polynomial time using an NP oracle). For example, consider the schema mapping $\mathcal{M}$ defined by the following GAV constraints:

$$\begin{array}{l}(\sigma_1)\ \forall x, y, z\ (S(x, y, z) \rightarrow T(x))\\ (\sigma_2)\ \forall x, y\ (S(x, y, y) \rightarrow T(x))\\ (\sigma_3)\ \forall x, y\ (S(x, x, x) \wedge S(x, y, x) \rightarrow V(x))\end{array}$$

This schema mapping is not normalized. It violates the first requirement because the canonical structure of σ_3 is not a core, and it violates the second requirement because there is a homomorphism from the canonical structure of σ_1 to the canonical structure of σ_2. A logically equivalent normalized schema mapping $\mathcal{M}'$ can be obtained from $\mathcal{M}$ by removing the conjunct $S(x, y, x)$ from σ_3 and by removing the entire constraint σ_2.

We call a GAV schema mapping *c-acyclic* if the canonical structure of each of its GAV constraints is c-acyclic.

Given a GAV constraint σ, a *join cycle* of σ is a sequence $x_1, F_1, x_2, F_2 \ldots, x_n F_n x_{n+1}$ ($n > 1$) where $x_1, x_2, \ldots$ are variables, $x_{n+1} = x_1$, each F_i is an atom from the left-hand-side of σ containing both x_i and x_{i+1}, and $F_i \neq F_{i+1}$ for all $i < n$ (this is to exclude trivial cycles traversing the same edge twice in opposite directions). An *exported variable* of σ is a variable occurring in the right-hand side of σ. Using these two notions, it is easy to see that c-acyclicity is equivalent to saying that the following two conditions hold:

- atoms may not contain two occurrences of the same non-exported variable
- each join cycle passes through an exported variable.

The schema mapping $\mathcal{M}$ described above is not c-acyclic, as the non-exported variable y of σ_2 occurs twice in the same conjunct. However, the normalized schema mapping $\mathcal{M}'$, where σ_2 is removed and the conjunct $S(x, y, x)$ removed from σ_3, is c-acyclic. The example schema mapping in Figure 1 is normalized and not c-acyclic.

From Theorem 9, we obtain the following characterization of the uniquely characterizable GAV schema mappings:

Theorem 10. *Every c-acyclic GAV schema mapping is uniquely characterizable, and a uniquely characterizing set of universal examples can be effectively computed from a given c-acyclic GAV schema mapping. Conversely, every uniquely characterizable GAV schema mapping $\mathcal{M}$ is logically equivalent to a c-acyclic GAV schema mapping; in fact, $\mathcal{M}$ is c-acyclic after normalization.*

It follows that, for instance, the schema mapping defined by the GAV constraint $\forall x_1, \ldots, x_n(S(x_1, x_2) \wedge \cdots \wedge S(x_{n-1}, x_n) \rightarrow T(x_1, x_n))$ is uniquely characterizable, as are schema mappings defined by GAV constraints whose variables are all exported.

In the remainder of this section, we analyze the complexity of various decision problems concerning unique characterizability and unique characterizations. In our complexity analysis, we assume that the source schema and target schema are fixed (and finite). This makes all reductions described in Section 3 polynomial time computable.

Corollary 3. *The following problem is* NP*-complete: given a GAV schema mapping, is it uniquely characterizable? If the schema mapping is normalized, the problem is in* LOGSPACE*.*

Corollary 4. *The following problem is decidable: given a GAV schema mapping $\mathcal{M}$ and a finite set of universal examples $\mathcal{U}$, does $\mathcal{U}$ uniquely characterize $\mathcal{M}$?*

Below, we will consider two additional decision problems.

Theorem 11. *The following problem is* DP*-complete: given a finite set of universal examples $\mathcal{U}$, is there a GAV schema mapping fitting $\mathcal{U}$? If the input consists of ground universal examples, the problem is* coNP*-complete. In both cases, the hardness holds already for a single universal example.*

Here, by a *ground* data example we mean a data example (I, J) such that the domain of J is a subset of the domain of I. The proof of Theorem 11, in effect, establishes something stronger: from any given finite set of universal examples $\mathcal{U}$, it is possible to compute in polynomial time a candidate GAV schema mapping $\mathcal{M}_{\mathcal{U}}$, such that if any GAV schema mapping fits the universal examples $\mathcal{U}$, then $\mathcal{M}_{\mathcal{U}}$ fits (in fact, $\mathcal{M}_{\mathcal{U}}$ is guaranteed to be the *logically weakest fitting schema mapping*, meaning that for any other schema mapping $\mathcal{M}'$ fitting the universal examples $\mathcal{U}$, the GAV constraints of $\mathcal{M}_{\mathcal{U}}$ logically imply those of $\mathcal{M}'$).

Theorem 12. *The following problem is decidable: given a finite set of universal examples $\mathcal{U}$, is there a unique schema mapping $\mathcal{M}$ that fits them?*

6 Concluding Remarks

We have established a tight connection between unique characterizability of GAV schema mappings via data examples, and homomorphism dualities, and we used this connection to obtain criteria and complexity results of unique characterizability for GAV schema mappings, and other related results.

The homomorphism dualities we considered in this paper consist of finite sets of structures. In the literature on constraint satisfaction problems, more general types of homomorphism dualities have been studied, for instance where one of the sets consists of infinitely many structures of bounded treewidth [10]. This raises the question whether known results about such dualities can be used to obtain further insights into the unique characterizability of schema mappings.

References

1. Fagin, R., Vardi, M.Y.: The Theory of Data Dependencies - A Survey. In: Proc. of Symposia in Applied Mathematics. Volume 34 - Mathematics of Information Processing. (1986) 19–71
2. Kolaitis, P.G.: Schema Mappings, Data Exchange, and Metadata Management. In: ACM Symposium on Principles of Database Systems (PODS). (2005) 61–75
3. Lenzerini, M.: Data Integration: A Theoretical Perspective. In: ACM Symposium on Principles of Database Systems (PODS). (2002) 233–246
4. Fagin, R., Kolaitis, P.G., Miller, R.J., Popa, L.: Data Exchange: Semantics and Query Answering. Theoretical Computer Science (TCS) **336**(1) (2005) 89–124
5. Barceló, P.: Logical foundations of relational data exchange. SIGMOD Record **38**(1) (2009) 49–58
6. Alexe, B., Kolaitis, P.G., Tan, W.C.: Characterizing schema mappings via data examples. In: ACM Symposium on Principles of Database Systems (PODS). (2010)
7. Hell, P., Nešetřil, J.: Graphs and Homomorphisms. Oxford University Press (2004)
8. Foniok, J., Nesetril, J., Tardif, C.: Generalised dualities and maximal finite antichains in the homomorphism order of relational structures. Eur. J. Comb. **29**(4) (2008) 881–899
9. ten Cate, B., Kolaitis, P.G.: Structural Characterizations of Schema-Mapping Languages. In: International Conference on Database Theory (ICDT). (2009) 63–72
10. Nešetřil, J., Zhu, X.: On bounded treewidth duality of graphs. Journal of Graph Theory **23**(2) (1998) 151 – 162
11. Hell, P., Nešetřil, J.: The core of a graph. Discrete Mathematics **109** (1992) 117–126
12. Nešetřil, J., Tardif, C.: Short answers to exponentially long questions: Extremal aspects of homomorphism duality. SIAM J. of Discrete Mathematics **19**(4) (2005) 914–920
13. Reingold, O.: Undirected st-connectivity in log-space. In: STOC ’05: Proceedings of the ACM symposium on Theory of Computing, ACM (2005) 376–385