Structural Characterizations of Schema-Mapping Languages*

Balder ten Cate†
University of Amsterdam and UC Santa Cruz
balder.tencate@uva.nl

Phokion G. Kolaitis
UC Santa Cruz and IBM Almaden
kolaitis@cs.ucsc.edu

ABSTRACT

Schema mappings are declarative specifications that describe the relationship between two database schemas. In recent years, there has been an extensive study of schema mappings and of their applications to several different data inter-operability tasks, including applications to data exchange and data integration. Schema mappings are expressed in some logical formalism that is typically a fragment of first-order logic or a fragment of second-order logic. These fragments are chosen because they possess certain desirable structural properties, such as existence of universal solutions or closure under target homomorphisms. In this paper, we turn the tables and focus on the following question: can we characterize the various schema-mapping languages in terms of structural properties possessed by the schema mappings specified in these languages? We obtain a number of characterizations of schema mappings specified by source-to-target (s-t) dependencies, including characterizations of schema mappings specified by LAV (local-as-view) s-t tgds, schema mappings specified by full s-t tgds, and schema mappings specified by arbitrary s-t tgds. These results shed light on schema-mapping languages from a new perspective and, more importantly, demarcate the properties of schema mappings that can be used to reason about them in data inter-operability applications.

Categories and Subject Descriptors
H.2.5 [Heterogeneous Databases]: Data translation
H.2.4 [Systems]: Relational databases

General Terms
Languages, Theory

Keywords
Schema mapping, data exchange, data integration, definability

1. INTRODUCTION

Schema mappings are declarative specifications that describe the relationship between two database schemas. In recent years, they have been used extensively in specifying and studying several different inter-operability tasks. In particular, schema mappings are regarded as the essential building blocks in data exchange and data integration (see, e.g., the surveys [10, 11]). The main task in data exchange is to transform data structured under one schema, called the source schema, into data structured under a different schema, called the target schema, in such a way that the constraints given by a schema mapping between the two schemas are satisfied. The main task in data integration is to answer queries against a global schema that is related to several heterogeneous local schemas via schema mappings.

Schema mappings are expressed in some logical language that is typically a restricted fragment of first-order logic or of second-order logic. The use of restricted fragments is dictated by the fact that the main algorithmic tasks in data exchange and data integration become undecidable if unrestricted use of even just first-order logic is allowed. Furthermore, the logical languages used to express schema mappings are chosen with two criteria in mind: on the one hand, they must be powerful enough to capture interesting specifications and, on the other, they must possess certain desirable structural properties. In data exchange, the prime example of such a desirable structural property is the existence of universal solutions, which, intuitively, are the preferred target instances to materialize [5]. This property is shared by both source-to-target tuple-generating dependencies (s-t tgds) [5] and by second-order tuple-generating dependencies (SO tgds) [7]. In data integration, an important example of such a desirable property is that the certain answers to a union of conjunctive queries over the target schema can be obtained by rewriting the query to a union of conjunctive queries over the source schema. One of the consequences of this rewriting property (which, like the existence of universal solutions, is possessed by both s-t tgds and SO tgds) is that the certain answers of unions of conjunctive queries over the target schema can be obtained in polynomial time. Two other examples of structural properties frequently used in the study of schema mappings are closure under target homomorphisms and closure under union. The former is a property shared by all s-t tgds, but not by all SO tgds. The latter is a property shared by all LAV (local-as-view) s-t tgds, but not by arbitrary s-t tgds; one of its many implications is that every schema mapping specified by LAV tgds has a quasi-inverse [8].

In this paper, we turn the tables and focus on the following question: can we characterize the various schema-mapping languages in terms of structural properties possessed by the schema mappings specified in these languages? We investigate this question and obtain a number of characterizations of schema mappings specified by s-t tgds, schema mappings specified by LAV (local-as-view) s-t tgds, and also schema mappings specified by full s-t tgds. Our char-

*Research on this paper partially supported by NSF Grant IIS-0430994.

†Work carried out during a research visit at UC Santa Cruz and the IBM Almaden Research Center; research of this author also supported by the Netherlands Organization for Scientific Research (NWO) grant 639.021.508.

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the ACM copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the ACM. To copy otherwise, or to republish, to post on servers or to redistribute to lists, requires a fee and/or special permissions from the publisher, ACM.
ICDT 2009, March 23–25, 2009, Saint Petersburg, Russia.
Copyright 2009 ACM 978-1-60558-423-2/09/0003 ...$5.00

acterizations are in terms of structural properties of schema mappings, including the ones mentioned above: existence of universal solutions, rewriting of unions of conjunctive queries over the target, closure under target homomorphisms, and closure under union. In particular, we establish that a schema mapping $\mathcal{M}$ is definable by a finite set of LAV tgds if and only if universal solutions w.r.t. $\mathcal{M}$ always exist, $\mathcal{M}$ allows for the rewriting of unions of conjunctive queries over the target, and $\mathcal{M}$ is closed under both target homomorphisms and union. In some cases, however, additional structural properties have to be singled out and used. Specifically, our characterization of schema mappings definable by s-t tgds involves a new property, which we call *n-modularity*: a schema mapping $\mathcal{M}$ is n-modular (where n is a positive integer) if whenever a target instance J is not a solution for some source instance I w.r.t. to $\mathcal{M}$, there is a sub-instance I' of I of size at most n such that J is a not solution for I' w.r.t. $\mathcal{M}$. With this property at hand, we establish that a schema mapping $\mathcal{M}$ is definable by a finite set of s-t tgds if and only if universal solutions w.r.t. $\mathcal{M}$ always exist, $\mathcal{M}$ allows for the rewriting of unions of conjunctive queries over the target, $\mathcal{M}$ is closed under target homomorphisms, and $\mathcal{M}$ is n-modular for some $n \geq 1$. Note that the aforementioned two results are characterizations of schema mappings without assuming that they are definable in some (richer) logic. We also obtain results of a different type in which we assume that the schema mappings are already definable in first-order logic and then characterize when they are definable by a finite set of s-t tgds or by a finite set of full s-t tgds. It should be pointed out that the proofs of the characterizations under the first-order definability assumption make essential use of Rossman's proof of the preservation-under-homomorphisms theorem [14].

The foremost motivation for carrying out this investigation is methodological. Specifically, our structural characterization theorems shed light on the exact set of tools available in the study of schema mappings specified in particular languages. To this effect, they demarcate the properties of schema mappings that can be used to reason about them in data inter-operability applications; furthermore, they pinpoint the properties of schema mappings that one loses or gains by switching from one schema-mapping language specification to another. As an application, we employ our structural characterization to derive complexity-theoretic results for testing definability of schema mappings in the schema-mapping languages considered here.

The remainder of this paper is organized as follows. In Section 2, we present the definitions of all the basic notions and give some simple results about the structural properties possessed by the various schema-mapping languages. In Section 3, we state and outline the proofs of our main structural characterization theorems for schema-mapping languages. In Section 4, we study the computational complexity of testing whether a schema mapping expressible in some schema-mapping language is also expressible in a different schema-mapping language. In Section 5, we briefly consider a different kind of expressive power, in the spirit of Bancilhon-Paredaens-completeness [3, 13], and address the question which target instances can be obtained from a source instance as universal solutions for schema mappings definable by a finite set of s-t tgds. Finally, in Section 6, we list some open problems and directions for future research.

2. PRELIMINARIES AND BASIC FACTS

2.1 Schema mappings, solutions, and certain answers

A *schema* is a tuple $\mathbf{R} = (R_1, \ldots, R_n)$ of relation symbols of fixed arities. An $\mathbf{R}$*-instance* is a tuple $I = (R_1^I, \ldots, R_n^I)$ of relations whose arities match those of the relation symbols of $\mathbf{R}$. A *fact* of I is an expression $R_i\mathbf{a}$ with $i \leq n$ and $\mathbf{a}$ a tuple of values belonging to the relation R_i^I. The *active domain* of I, denoted by $adom(I)$, is the set of all values occurring in the relations R_i^I, for $1 \leq i \leq n$. Unless explicitly stated otherwise, we will always restrict ourselves to finite instances I, i.e., $adom(I)$ is a finite set. We will be working with two disjoint schemas, called the *source schema* $\mathbf{S} = (S_1, \ldots, S_n)$ and the *target schema* $\mathbf{T} = (T_1, \ldots, T_m)$. An $\mathbf{S}$-instance is called a *source instance* and an $\mathbf{T}$-instance is called a *target instance*. Whenever we consider a pair of instances (I, J), it will be implicitly understood that I is a source instance and J is a target instance.

A *schema mapping* is a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \mathcal{W})$, where $\mathbf{S}$ is a source schema, $\mathbf{T}$ is a target schema disjoint from $\mathbf{S}$, and $\mathcal{W}$ is a class of pairs (I, J) of $\mathbf{S}$-instances and $\mathbf{T}$-instances that is closed under isomorphisms, i.e., if $(I, J) \in \mathcal{W}$ and $h : (I \cup J) \cong (I' \cup J')$ is an isomorphism, then $(I', J') \in \mathcal{W}$. If a pair $(I, J) \in \mathcal{W}$, then we say that J is a *solution for I w.r.t. $\mathcal{M}$* (or, simply, a *solution for I* whenever $\mathcal{M}$ is understood from the context). For each source instance I, we write $\mathrm{Sol}_{\mathcal{M}}(I)$ to denote the set of all solutions of I w.r.t. $\mathcal{M}$.

Let L be a logical language and let Σ be a set of L-sentences. We say that a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \mathcal{W})$ is *L-definable by* Σ if $(I, J) \in \mathcal{W}$ if and only if $(I, J) \models \Sigma$. In this case, we will often identify the schema mapping $\mathcal{M}$ with the triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$. Note that, since we are restricting ourselves to finite instances, L-definability means L-definability within the class of (pairs of) finite instances. Here, we are mainly interested in definability in first-order logic (FO-definability), as well as in definability in fragments of first-order logic that are used as specification languages in data exchange and data integration. Note that if the source schema $\mathbf{S}$ and the target schema $\mathbf{T}$ are understood from the context, we will often identify a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \mathcal{W})$ with the class $\mathcal{W}$ of pairs of instances or with a set Σ of L-sentences that defines it. We will also often write $(I, J) \in \mathcal{M}$, instead of $(I, J) \in \mathcal{W}$.

If q is a k-ary query over the target and I is a source instance, then the *certain answers of q on I w.r.t. $\mathcal{M}$* is the set $\mathsf{certain}_{\mathcal{M}}(q)(I) = \bigcap\{q(J) \mid J \in Sol_{\mathcal{M}}(I)\}$. It is easy to see that every tuple in $\mathsf{certain}_{\mathcal{M}}(q)(I)$ is a tuple of values from I. Thus, every schema mapping $\mathcal{M}$ induces a transformation $\mathsf{certain}_{\mathcal{M}}$ from queries over the target schema to queries over the source schema, so that if q is a k-ary query over the target schema, then this transformation produces the query $\mathsf{certain}_{\mathcal{M}}(q)$ over the source schema defined by $\mathsf{certain}_{\mathcal{M}}(q)(I) = \bigcap\{q(J) \mid J \in Sol_{\mathcal{M}}(I)\}$.

2.2 Homomorphisms, conjunctive queries, and universal solutions

A central notion in the study of schema mappings is that of a *homomorphism*. If K and K' are two instances over the same schema $\mathbf{R} = (R_1, \ldots, R_n)$, then a *homomorphism* from K to K' is a function from the active domain of K to the active domain of K' such that if $(a_1, \ldots, a_m) \in R_i^K$, then $(h(a_1, \ldots, a_m)) \in R_i^{K'}$, for $i = 1, \ldots, n$. If there is a homomorphism from K to K', then we say that K' is a *homomorphic extension* of K, and that K is *homomorphically embedded* in K'. If there are homomorphisms $h : K \to K'$ and $h' : K' \to K$, then we say that K and K' are *homomorphically equivalent*. A homomorphism h is said to be *constant* on a set X if h restricted to $X \cap dom(h)$ is the identity function on this set. Often, when we will consider homomorphisms between target instances, we will need to require that they are constant on the domain of some source instance.

A k-ary query q is said to be *preserved under homomorphisms* if

for all homomorphisms $h : K \to K'$ and all tuples $(d_1, \ldots, d_k) \in q(K)$, we have that $(h(d_1), \ldots, h(d_k)) \in q(K')$. A *positive query* is a query defined by a positive existential FO-formula or, equivalently, a query that is a union of conjunctive queries. All positive queries are preserved under homomorphisms.

We now recall the definition of universal solutions from [5]. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \mathcal{W})$ be a schema mapping and let I be a source instance. A target instance J is a *universal solution for I* if J is a solution for I and for each target instance J' that is a solution for I, there is a homomorphism $h : J \to J'$ that is constant on $adom(I)$.

2.3 Schema mapping languages

DEFINITION 2.1. *Let* $\mathbf{S}$ *be a source schema and* $\mathbf{T}$ *a target schema.*

1. *A* source-to-target tuple-generating dependency (s-t tgd) *is a FO-formula of the form*

$$\forall \mathbf{x}(\phi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})),$$

where $\phi(\mathbf{x})$ is a conjunction of atomic formulas over $\mathbf{S}$ with variables in $\mathbf{x}$, each variable in $\mathbf{x}$ occurs in at least one atomic formula in $\phi(\mathbf{x})$, and $\psi(\mathbf{x}, \mathbf{y})$ is a conjunction of atomic formulas over $\mathbf{T}$ with variables in $\mathbf{x}$ and $\mathbf{y}$.

2. *A* full s-t tgd *is a s-t tgd with no existential quantifiers in the right-hand side, i.e., it is of the form $\forall \mathbf{x}(\phi(\mathbf{x}) \to \psi(\mathbf{x}))$, where $\phi(\mathbf{x})$ and $\psi(\mathbf{x})$ are conjunctions of atomic formulas over $\mathbf{S}$ and $\mathbf{T}$ respectively, and each variable in $\mathbf{x}$ occurs in at least one atomic formula in $\phi(\mathbf{x})$.*

3. *A* LAV (local-as-view) s-t tgd *is a s-t tgd in which the left-hand side is a single atomic formula (we do not assume that each variable from $\mathbf{x}$ occurs only once in the atomic formula).*

We will focus on schema mappings that are definable by finite sets of s-t tgds, by finite sets of full s-t tgds, and by finite sets of LAV s-t tgds. These classes of schema mappings have been investigated extensively in data exchange and data integration. We note that s-t tgds are also known in the literature as GLAV (global-and-local-as-view) constraints under sound view semantics (see [11]). Moreover, every full s-t tgd is equivalent to a finite set of GAV constraints, which are the special case of full s-t tgds in which the right-hand side has a single atomic formula.

2.4 Structural properties of schema mappings

In this section, we present several structural properties of schema mappings that will play a key role in our characterizations. We begin with three such properties that have been used in both data exchange and data integration.

DEFINITION 2.2. *Let $\mathcal{M}$ be a schema mapping.*

- ***Closure under target homomorphisms:*** *We say that $\mathcal{M}$* is closed under target homomorphisms *if for all $(I, J) \in \mathcal{M}$ and for all homomorphisms $h : J \to J'$ that are constant on $adom(I)$, we have that $(I, J') \in \mathcal{M}$.*
- ***Admitting universal solutions:*** *We say that $\mathcal{M}$* admits universal solutions *if for each source instance I there is a universal solution J for I.*
- ***Allowing for conjunctive query rewriting:*** *We say that $\mathcal{M}$* allows for conjunctive query rewriting *if for each union q of conjunctive queries over the target schema, the certain answers query $\mathsf{certain}_{\mathcal{M}}(q)$ is definable by a union of conjunctive queries over the source schema. (Here, as it is usually the case, "union of conjunctive queries" means a finite union of conjunctive queries, and equalities are allowed in the conjunctive queries)*

The first two conditions of closure under target homomorphisms and admitting universal solutions go very well together. As was observed in [5], if a schema mapping is closed under target homomorphisms and admits universal solutions, then, for every source instance I, the (typically infinite) space $Sol_{\mathcal{M}}(I)$ of all solutions of I can be completely described by a single target instance J, namely, by any universal solution J for I. This is so because if J is universal for I and $\mathcal{M}$ is closed under target homomorphisms, then for every target instance J', we have that J' is a solution for I if and only if there is a homomorphism $h : J \to J'$ that is constant on $adom(I)$. Thus, these two conditions lie at the foundation of *data exchange*. The third condition of allowing for conjunctive query rewriting is important in the context of *data integration*, since it implies that the certain answers of unions of conjunctive queries over the target are computable in polynomial time (in the sense of data complexity).

It is well known that all three conditions of closure under target homomorphisms, admitting universal solutions, and allowing for conjunctive query rewriting are possessed by every schema mapping $\mathcal{M}$ definable by a finite set of s-t tgds. Closure under homomorphisms follows easily from the definitions; admitting universal solutions was shown in [5] using the *chase procedure*. In the case of full s-t tgds, a union of conjunctive queries over the target is easily transformed to a union of conjunctive queries over the source by simply replacing each target relation symbol P by a union of conjunctive queries over the source that defines P. In the case of arbitrary s-t tgds, allowing for conjunctive query rewriting is proved by first "decomposing" the given s-t tgds to full s-t tgds and to LAV s-t tgds, and then applying results from [1] and [4]. We collect these facts into one proposition.

PROPOSITION 2.3. *Every schema mapping definable by a finite set of s-t tgds is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting.*

In the full version of this paper, we show that even schema mappings speficied by a *second-order tgd* (SO tgd) (see [7] for the definition) allows for conjunctive query rewriting; since a finite set of SO tgds is known to be equivalent to a single SO tgd, this rewriting result holds also for finite sets of SO tgds. Moreover, we show that the query rewriting can be performed in polynomial time (measured in the combined size of the schema mapping specification and the conjunctive query) if the output query is allowed to be presented in the form of a positive existential first-order formula and the domain of the source instance contains at least two distinguished constants. It follows that the combined complexity of query answering (where the input is the SO tgd, the source instance, and the target conjunctive query) is NP-complete. Since s-t tgds can be translated into SO tgds in linear time, the same holds for schema mappings specified by finite sets of s-t tgds. These two results, stated below, are used in Section 3.4 and in Lemma 4.1.

PROPOSITION 2.4. *Every schema mapping definable by a SO tgd allows for conjunctive query rewriting.*

PROPOSITION 2.5. *The following problem is NP-complete:*
Given a schema mapping $\mathcal{M}$ specified by a SO tgd, a source instance I, a k-ary positive target query q $(k \geq 0)$ and a k-tuple $\mathbf{a}$ of elements from $adom(I)$, does $\mathbf{a}$ belong to $\mathsf{certain}_{\mathcal{M}}(q)(I)$?

Next, we define three additional properties of schema mappings.

DEFINITION 2.6. *Let $\mathcal{M}$ be a schema mapping.*

- ***Closure under target intersection:*** *We say that $\mathcal{M}$* is closed under target intersection *if for all source instances I and all target instances J_1, J_2, if $(I, J_1) \in \mathcal{M}$ and $(I, J_2) \in \mathcal{M}$, then also $(I, J_1 \cap J_2) \in \mathcal{M}$.*
- ***Closure under union:*** *We say that $\mathcal{M}$ is* closed under union *if $(\emptyset, \emptyset) \in \mathcal{M}$, and for all $(I, J) \in \mathcal{M}$ and $(I', J') \in \mathcal{M}$ (not necessarily disjoint), we have that also $(I \cup I', J \cup J') \in \mathcal{M}$.*

  Intuitively, a schema mapping is closed under union if solutions can be constructed in a "modular" fashion, i.e., on a tuple-by-tuple basis.
- ***n-Modularity:*** *Let n be a positive integer. We say that $\mathcal{M}$ is n-*modular *if whenever a pair (I, J) does not belong to $\mathcal{M}$, there is a sub-instance $I' \subseteq I$ such that $|adom(I')| \leq n$ and (I', J) does not belong to $\mathcal{M}$.*

  Intuitively, n-modularity asserts that if $(I, J) \notin \mathcal{M}$, then there is a concise explanation for this fact; this property can also be viewed as a relaxation of closure under union.
- ***Reflecting source homomorphisms:*** *We say that $\mathcal{M}$* reflects source homomorphisms *if for all source instances I, I' and for all target instances J, J' such that J is a universal solution for I and J' is a solution for I', we have that every homomorphism $h : I \to I'$ extends to a homomorphism from J to J'.*

  Note that, in this definition, we do not require the homomorphisms to be constant on $adom(I)$.

We now give several useful propositions about the properties we just introduced.

PROPOSITION 2.7. *Let $\mathcal{M}$ be a schema mapping.*

- *If $\mathcal{M}$ is definable by a finite set of full s-t tgds, then $\mathcal{M}$ is closed under target intersection.*
- *If $\mathcal{M}$ is definable by a finite set of LAV s-t tgds, then $\mathcal{M}$ is closed under union.*
- *If $\mathcal{M}$ is definable by a finite set of s-t tgds, then $\mathcal{M}$ is n-modular for some $n \geq 1$.*

PROOF. The first two parts follow easily from the definitions. For the third part, assume that $\mathcal{M}$ is a schema mapping definable by a finite set Σ of s-t tgds. Let n be the maximum number of variables occurring in the left-hand side of the s-t tgds in Σ. We claim that $\mathcal{M}$ is n-modular. Assume that $(I, J) \notin \mathcal{M}$. Then there is an s-t tgd

$$\forall \mathbf{x}(\phi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}))$$

from Σ and a tuple $\mathbf{a}$ of values from $adom(I)$ such that $(I, J) \models \phi(\mathbf{a}) \wedge \neg \exists \mathbf{y} \psi(\mathbf{a}, \mathbf{y})$. Now, let I' be the sub-instance of I containing only the values $\mathbf{a}$. Then it is still the case that $(I', J) \models \phi(\mathbf{a}) \wedge \neg \exists \mathbf{y} \psi(\mathbf{a}, \mathbf{y})$, and hence $(I', J) \notin \mathcal{M}$. $\square$

PROPOSITION 2.8. *If a schema mapping $\mathcal{M}$ is closed under target homomorphisms and target intersections, then for every source instance I and every target instance J, we have that J is a solution for I if and only if $J \restriction adom(I)$ is a solution for I, where $J \restriction adom(I)$ is the sub-instance of J consisting of all tuples with values from $adom(I)$ only.*

PROOF. One direction follows immediately from the fact that the inclusion map from $J \restriction adom(I)$ into J is a homomorphism. For the other direction, suppose that J is a solution for I. Let J' be an isomorphic copy of J agreeing with J on elements in $adom(I)$, but disjoint for the rest of the active domain. Then, by closure under isomorphisms, J' is a solution for I, hence $J' \cap J$ is a solution for I by closure under intersection. By construction, $J \cap J' = J \restriction adom(I)$. $\square$

PROPOSITION 2.9. *If $\mathcal{M}$ is a schema mapping that is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting, then $\mathcal{M}$ reflects source homomorphisms.*

PROOF. Let J be a universal solution for I, let J' be a solution for I', and let $h : I \to I'$ be a homomorphism. For each element of I, choose a first-order variable and form the canonical conjunctive query q of J in these free variables (thus only the elements of $adom(J) \setminus adom(I)$ are existentially quantified in q). Clearly, q is true in J under the natural assignment that sends each variable to the corresponding element of I. Since J is a universal solution for I, we have that q is true in every other solution of I as well, under the same assignment. Hence, $\mathsf{certain}_{\mathcal{M}}(q)$ is true in I under this assignment. Since $\mathcal{M}$ allows for conjunctive query rewriting, $\mathsf{certain}_{\mathcal{M}}(q)$ is definable by a union of conjunctive queries over the source schema, hence it is preserved by the homomorphism h. This means that $\mathsf{certain}_{\mathcal{M}}(q)$ is true in I' under the assignment that sends every variable to the h-image of the corresponding element of I, and hence $q(J')$ is true under this assignment as well. In other words, there is a a homomorphism from J to J' that extends q. $\square$

Note that the proof of the above Proposition 2.9 relies heavily on the assumption that we consider finite instances only.

3. CHARACTERIZATIONS OF SCHEMA-MAPPING LANGUAGES

3.1 LAV s-t tgds

The following provides a characterization of LAV s-t tgds.

THEOREM 3.1. *For all schema mappings $\mathcal{M}$, the following are equivalent:*

1. *$\mathcal{M}$ is definable by a finite set of LAV s-t tgds*
2. *$\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, allows for conjunctive query rewriting, and is closed under union.*
3. *$\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, reflects source homomorphisms, and is closed under union.*

PROOF. The implications $(1) \Rightarrow (2)$ and $(2) \Rightarrow (3)$ are proved in Section 2. We prove $(3) \Rightarrow (1)$. The idea behind the proof is as follows: since $\mathcal{M}$ is closed under union, universal solutions for source instances I can be constructed out of universal solutions for parts of I. This implies that, in defining our schema mapping, we only need to take into account a finite number of source instances up to isomorphism, namely, those that contain precisely one tuple. In what follows we will make this idea precise.

Suppose $\mathcal{M}$ satisfies the listed conditions. Let $R_1, \ldots, R_n$ be the relations of the source schema, and let D be a set consisting of k distinct values, with $k = \max_{i \leq n} \mathsf{arity}(R_i)$. Let facts be the set of all possible facts, of the form $R_i(d_1, \ldots, d_\ell)$ with $i \leq n$, $\ell = \mathsf{arity}(R_i)$, and $d_1, \ldots, d_\ell \in D$. For each $\alpha \in \mathsf{facts}$, let I_α be the source instance containing precisely one fact, namely α, and let J_α be a universal solution for I_α. Let $PosDiag_{I_\alpha}(\mathbf{x})$ be the positive diagram of I_α, i.e., the set of all facts true in I (which consists of precisely one fact) and let $PosDiag_{J_\alpha}(\mathbf{x}, \mathbf{y})$ be the positive diagram of J_α, where $\mathbf{x}$ are as many variables as there

are elements of $adom(I_\alpha)$ and $\mathbf{y}$ as many variables as there are elements of $adom(J_\alpha) \setminus adom(I_\alpha)$. Define ϕ_α to be the following LAV s-t tgd:

$$\forall \mathbf{x}(PosDiag_{I_\alpha}(\mathbf{x}) \rightarrow \exists \mathbf{y}(PosDiag_{J_\alpha}(\mathbf{x}, \mathbf{y})))$$

Finally, let $\Sigma = \{\phi_\alpha \mid \alpha \in \mathsf{facts}\}$. We claim that Σ defines $\mathcal{M}$.

First, we prove soundness: every $(I, J) \in \mathcal{M}$ satisfies Σ. Suppose $(I, J) \in \mathcal{M}$, and take any $\phi_\alpha \in \Sigma$. Furthermore, suppose that the antecedent of ϕ_α is satisfied in (I, J) under some variable assignment h. In other words, h is a homomorphism from I_α to I. Since $\mathcal{M}$ reflects source homomorphisms, there is a homomorphism $h' : J_\alpha \rightarrow J$ extending h. This means precisely that the consequent of ϕ_α is satisfied in J under the assignment h'. Hence, (I, J) satisfies ϕ_α.

Next, we prove completeness: every pair (I, J) satisfying Σ belongs to $\mathcal{M}$. Suppose (I, J) satisfies Σ. If $I = \emptyset$ then $(I, \emptyset) \in \mathcal{M}$ by definition of closure under union, and hence, by closure under target homomorphisms, $(I, J) \in \mathcal{M}$. If not, let $I = I_1 \cup \ldots \cup I_n$ where each I_i contains only a single fact. Then each (I_i, J) still satisfies Σ. Since I_i contains a single fact, it must be isomorphic to I_α for some $\alpha \in \mathsf{facts}$. Using the fact that (I_i, J) satisfies ϕ_α, we can show that there is a homomorphism from a universal solution of I_i to J, constant on $adom(I_i)$, hence, by closure under target homomorphisms, $(I_i, J) \in \mathcal{M}$. It follows by closure under union that $(I, J) \in \mathcal{M}$. $\square$

Theorem 3.1 implies in particular that every schema mapping satisfying the conditions listed in (2) or (3) is definable by a first-order sentence (or, equivalently, by a finite set of first-order sentences).

3.2 Full s-t tgds

The following provides a characterization of schema mappings definable by a finite set of full s-t tgds:

THEOREM 3.2. *For all schema mappings $\mathcal{M}$, the following are equivalent:*

1. *$\mathcal{M}$ is definable by a finite set of full s-t tgds*
2. *$\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, allows for conjunctive query rewriting, and is closed under target intersection.*
3. *$\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, reflects source homomorphisms, is closed under target intersection, and is n-modular for some $n \geq 1$.*

PROOF. (Sketch) The implication (1) $\Rightarrow$ (2) is proved in Section 2.

For the implication (2) $\Rightarrow$ (3), suppose $\mathcal{M}$ satisfies the conditions listed under (2). We need to show that $\mathcal{M}$ reflects source homomorphisms and is n-modular for some $n > 0$. That $\mathcal{M}$ reflects source homomorphisms follows from Proposition 2.9. Next, for each target relation R, let $q_R = \mathsf{certain}_{\mathcal{M}}(R\mathbf{y})$, where $\mathbf{y}$ is a sequence of distinct fresh variables, of appropriate length. Note that, since $\mathcal{M}$ allows for conjunctive query rewriting, q_R can be written as a union of conjunctive queries. Now, let n be the maximum of the number of variables occurring in each q_R. We claim that $\mathcal{M}$ is n-modular. To see this, let I, J be any source and target instance such that $(I, J) \notin \mathcal{M}$. By Proposition 2.8, we may assume without loss of generality that $adom(J) \subseteq adom(I)$. Let J' be a universal solution for I with respect to $\mathcal{M}$. Since J is not a solution for I, it is not a homomorphic extension of J', and hence there is a tuple $\mathbf{d}$ that belongs to some relation R in J' but not in J. It follows that $\mathbf{d} \in q_R(I)$. Now, let I' be a sub-instance of I containing just enough elements to witness the existential quantifiers of q_R, so that $\mathbf{d} \in q_R(I')$. Then, $|adom(I)| \leq n$ and J is not a solution for I. This shows that $\mathcal{M}$ is n-modular.

The implication (3) $\Rightarrow$ (1) is established along the same lines as the proof of Theorem 3.1. Instead of considering all source instances consisting of one tuple, we consider all source instances I with $|adom(I)| \leq n$. There are only finitely many such source instances, up to isomorphism. Moreover, by Proposition 2.8, each has a null-free universal solution, and hence only *full* s-t tgds are needed to describe them. $\square$

In the third clause of Theorem 3.2, the requirement of n-modularity is necessary, as shown next:

EXAMPLE 3.3. *The following schema mapping, defined by an infinite set of full s-t tgds, satisfies all closure conditions listed in the third clause of Theorem 3.2 except n-modularity for any $n > 0$, and is not definable by any finite set of s-t tgds:*

$$\{\forall x_1, \ldots, x_n(Rx_1x_2 \wedge \cdots \wedge Rx_{n-1}x_n \rightarrow Sx_1x_n) \mid n > 0\}$$

It defines the class of all pairs (I, J) where S^J contains the transitive closure of R^I. Note that this schema mapping does not allow for conjunctive query rewriting: $\mathsf{certain}_{\mathcal{M}}(Sxy)$ is not definable by a union of conjunctive queries.

We note that an easy adaptation of the proof of Theorem 3.2 shows that schema mappings satisfying all conditions except n-modularity are still definable by an *infinite* set of full s-t tgds.

For schema mappings definable by a FO-sentence (unlike the above example), the requirement of n-modularity *can* be dropped.

THEOREM 3.4. *For all schema mappings $\mathcal{M}$ definable by a FO-sentence, the following are equivalent:*

1. *$\mathcal{M}$ is definable by a finite set of full s-t tgds*
2. *$\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, reflects source homomorphisms, and is closed under target intersection.*

The proof of Theorem 3.4 is considerably more involved than that of Theorem 3.2. In particular, it uses the following lemma, due to Rossman [14], which lies at the heart of his homomorphism preservation theorem in the finite.

LEMMA 3.5 ([14]). *For each $k > 0$ there is an $\ell > 0$ such that the following holds for all instances K_1 and K_2 of the same schema. If all Boolean positive queries of quantifier rank at most ℓ true in K_1 are true in K_2, then there are instances $\widehat{K_1}, \widehat{K_2}$ such that*

- *K_1 and $\widehat{K_1}$ are homomorphically equivalent,*
- *$\widehat{K_1}$ and $\widehat{K_2}$ satisfy the same FO sentences up to quantifier rank k, and*
- *There is a homomorphism from $\widehat{K_2}$ to K_2.*

PROOF OF THEOREM 3.4. We prove the implication (2) $\Rightarrow$ (1): let $\mathcal{M}$ be a schema mapping definable by a FO-sentence, satisfying the listed conditions. First, we will show that these together imply another closure property. Let the *target-inverted joint schema* be the schema containing all relations from the source schema, plus a relation $\bar{R}$ for each relation R of the target schema (of equal arity). The idea is that $\bar{R}$ represents the complement of R. Each pair (I, J) gives rise to a single instance over the target-inverted joint schema, denoted by $I \oplus \bar{J}$:

$R^{I \oplus \bar{J}} = R^I$ for a source relation R

$\bar{S}^{I \oplus \bar{J}} = (adom(I) \cup adom(J))^k \setminus S^J$ for a target relation S of arity k

The extra closure property we can infer now reads as follows:

Claim 1: For all pairs (I, J) and (I', J'), if there is a homomorphism from $I \oplus \bar{J}$ to $I' \oplus \bar{J'}$ and $(I', J') \in \mathcal{M}$ then $(I, J) \in \mathcal{M}$.

Proof of claim: By Proposition 2.8, we may assume without loss of generality that $adom(J) \subseteq adom(I)$ and $adom(J') \subseteq adom(I')$. Every homomorphism $h : I \oplus \bar{J} \to I' \oplus \bar{J'}$ is in particular a homomorphism from I to I'. Since $(I', J') \in \mathcal{M}$ and $\mathcal{M}$ reflects source homomorphisms, we have that h extends to a homomorphism $\widehat{h} : J'' \to J'$ for some universal solution J'' of I. We may again assume without loss of generality that $adom(J'') \subseteq adom(I)$, and hence $\widehat{h} = h$. Now, the identity function is a homomorphism from J'' to J, and hence $(I, J) \in \mathcal{M}$ by closure under target homomorphisms. To see that the identity function is a homomorphism from J'' to J, note that if $\mathbf{d} \in R^{J''}$, then, since $h : J'' \to J$, $h(\mathbf{d}) \in R^{J'}$, and hence, since $h : I \oplus \bar{J} \to I \oplus \bar{J'}$ preserves inverted target relations, we have that $\mathbf{d} \in R^J$.

Next, we will apply Lemma 3.5. Let k be the quantifier rank of any FO-sentence ϕ defining $\mathcal{M}$, and let ℓ be as described in Lemma 3.5. Furthermore, let m be ℓ multiplied by the number of distinct positive existential FO-sentences with quantifier depth at most ℓ over the target-inverted joint schema, containing only (inverted) relation symbols occurring in ϕ. Let $Th_m(\mathcal{M})$ be the (finite) set of all full s-t tgds of quantifier rank at most m that hold in all pairs $(I, J) \in \mathcal{M}$. We will show that all pairs (I, J) satisfying $Th_m(\mathcal{M})$ belong to $\mathcal{M}$, and hence $Th_m(\mathcal{M})$ *defines* $\mathcal{M}$.

Let $(I, J) \models Th_m(\mathcal{M})$. By Proposition 2.8 and the closure under target homomorphisms, we may assume without loss of generality that $adom(J) \subseteq adom(I)$.

Claim 2: For each $X \subseteq adom(I)$ with $|X| \leq m$, $((I \restriction X), (J \restriction X)) \in \mathcal{M}$.

Proof of claim: Introduce variables $\mathbf{x}$ for the elements of X. Let $PosDiag(I, X)$ be the set of all atomic formulas in these variables true in I, and let $NegDiag(J, X)$ be the set of all atomic formulas in these variables false in J.

There are two cases: if $NegDiag(J, X) = \emptyset$, then, using the fact that $\mathcal{M}$ admits solutions, it can be shown that $(I \restriction X, J \restriction X) \in \mathcal{M}$. In what follows, we therefore assume that $NegDiag(J, X) \neq \emptyset$.

Consider any $\chi \in NegDiag(J, X)$. First, we show that there is a pair $(I', J') \in \mathcal{M}$ satisfying $PosDiag(I, X) \cup \{\neg\chi\}$. For, if not, then $\forall \mathbf{x}(\bigwedge PosDiag(I, X) \to \chi) \in Th_m(\mathcal{M})$, where $\mathbf{x}$ is an enumeration of the elements of X. This would contradict the fact that $(I, J) \models Th_m(\mathcal{M})$. We can therefore conclude that there is a pair $(I', J') \in \mathcal{M}$ satisfying $PosDiag(I, X) \cup \{\neg\chi\}$. Moreover, by construction, there is a homomorphism $h : (I \restriction X) \to I'$. Now, let J'' be any universal solution for $I \restriction X$ with respect to $\mathcal{M}$. We may again assume without loss of generality that $adom(J'') \subseteq X$. Since $\mathcal{M}$ reflects source homomorphisms, h is a homomorphism from J'' to J'. In particular, $J'' \not\models \chi$. Since this holds for each $\chi \in NegDiag(J, X)$, we have that J'' is contained in $J \restriction X$, and hence, since $\mathcal{M}$ is closed under target homomorphisms, $((I \restriction X), (J \restriction X)) \in \mathcal{M}$.

Claim 3: There is a pair $(I', J') \in \mathcal{M}$ such that for every positive existential FO-sentence ϕ of quantifier rank at most ℓ over the target-inverted joint schema, if $I \oplus \bar{J} \models \phi$ then $I' \oplus \bar{J'} \models \phi$.

Proof of claim: For each distinct positive existential FO-sentence ϕ of quantifier rank at most ℓ over the target-inverted joint schema true in $I\bar{J}$, pick a witnessing subset $X \subseteq adom(I)$, with $|X| \leq \ell$. Take the union of all these sets X for the different sentences ϕ. The result is a subset of $adom(I)$ size at most m. Finally, apply Claim 2 on this set. *End of proof of claim*

It follows by Lemma 3.5 that there are pairs $\widehat{I \oplus \bar{J}}$ and $\widehat{I' \oplus \bar{J'}}$ such that $I \oplus \bar{J}$ and $\widehat{I \oplus \bar{J}}$ are homomorphically equivalent, $\widehat{I \oplus \bar{J}}$ and $\widehat{I' \oplus \bar{J'}}$ satisfy the same FO-sentences up to quantifier rank k, and there is a homomorphism from $\widehat{I' \oplus \bar{J'}}$ to $I' \oplus \bar{J'}$. We can now chase the diagram: $(I', J') \in \mathcal{M}$, and hence, by Claim 1, $(\widehat{I'}, \widehat{J'}) \in \mathcal{M}$; since $\mathcal{M}$ is defined by an FO-sentence of quantifier depth k, it follows that $(\widehat{I}, \widehat{J}) \in \mathcal{M}$, hence, again by Claim 1, $(I, J) \in \mathcal{M}$. $\square$

3.3 Arbitrary s-t tgds

As we saw in Section 2, every schema mapping defined by a *finite* set of s-t tgds is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting. Conversely, any schema mapping satisfying these conditions is definable by an *infinite* set of s-t tgds:

PROPOSITION 3.6. *If a schema mapping $\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting, then $\mathcal{M}$ is definable by an infinite set of s-t tgds.*

PROOF. (Sketch) Let $\mathcal{M}$ satisfy the listed properties. Consider any source instance I and target instance J such that J is a universal solution for I with respect to $\mathcal{M}$. For each element of $adom(I)$, introduce a distinct variable x_i, and for each element of $adom(J) \setminus adom(I)$, introduce a distinct variable y_j. Define $PosDiag_I(\mathbf{x})$ to be the set of all atomic formulas in $\mathbf{x}$ true in I (under the chosen assignment) and define $PosDiag_J(\mathbf{x}, \mathbf{y})$ likewise. Finally, let Σ be the set of all s-t tgds of the form

$$\phi_{I,J} \;:=\; \forall \mathbf{x}(\bigwedge PosDiag_I(\mathbf{x}) \to \exists \mathbf{y}(\bigwedge PosDiag_J(\mathbf{x}, \mathbf{y})))$$

where I is any source instance and J a universal solution for I with respect to $\mathcal{M}$. To ensure Σ is a set, not a proper class, we consider each source instance only once up to isomorphism. It can be shown that Σ defines $\mathcal{M}$, using an argument analogous to the one used in the proof of Theorem 3.1. $\square$

The following two examples show that Proposition 3.6 cannot be turned into a characterization of definability by *finite* sets of s-t tgds, nor a characterization of definability by *infinite* sets of s-t tgds.

EXAMPLE 3.7. *The schema mapping defined by the FO-sentence $\forall x \exists y \forall z (Rxz \to Syz)$ is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting, but is not definable by a finite set of s-t tgds [7, Proposition 3.4].*

EXAMPLE 3.8. *The schema mapping defined by the infinite set of s-t tgds*

$$\{\forall x(Px \to \exists y_1 \ldots y_n(Rxy_1 \wedge Ry_1y_2 \wedge \cdots \wedge Ry_{n-1}y_n)) \mid n \geq 0\}$$

does not admit (finite) universal solutions: it is quite easy to see that no finite solution for $I = \{Pa\}$ can be universal.

Thus, additional properties must be considered in order to characterize the schema mappings that are definable by a finite set of s-t tgds. It turns out that adding n-modularity as a requirement yields such a characterization.

THEOREM 3.9. *For all schema mappings $\mathcal{M}$, the following are equivalent:*

1. *$\mathcal{M}$ is definable by a finite set of s-t tgds*
2. *$\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, allows for conjunctive query rewriting, and is n-modular for some $n > 0$.*
3. *$\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, reflects source homomorphisms, and is n-modular for some $n > 0$.*

PROOF. (Hint) Along the same lines as the proof of Theorem 3.1 and Theorem 3.2. □

An alternative characterization can be obtained by using a property known as *bounded fact-block size* [6].

DEFINITION 3.10. *Let J is a solution for a source instance I with respect to some schema mapping $\mathcal{M}$.*

The fact graph *of J is the graph whose vertices are atomic facts of J and where two facts are connected by an edge is they share a null (i.e., a value not from the domain of I).*

A fact block *of J is a connected component of the fact graph.*

By the size *of a fact block we will mean the number of facts it contains.*

We say that a schema mapping $\mathcal{M}$ has bounded fact-block size *if for some $n > 0$, each source instance has a universal solution whose fact blocks have size at most n.*[1]

It was shown in [6] that schema mappings definable by a finite set of s-t tgds have bounded fact block size.

THEOREM 3.11. *For all schema mappings $\mathcal{M}$, the following are equivalent:*

1. *$\mathcal{M}$ is definable by a finite set of s-t tgds*
2. *$\mathcal{M}$ is closed under target homomorphisms, admits universal solutions, allows for conjunctive query rewriting, and has bounded fact-block size.*

PROOF. (Sketch.) We prove the implication (2) $\Rightarrow$ (1). Let $\mathcal{M}$ satisfy the listed conditions. In particular, let $n > 0$ be such that every source instance has a universal solution with respect to $\mathcal{M}$ whose fact blocks are of size at most n. Since there are only finitely many isomorphism types of fact blocks of size at most n, there is a finite set of isomorphism types of fact blocks (or, "fact-block types"), such that every source instance has a universal solution in which every fact block is of one of these types. Since $\mathcal{M}$ allows for conjunctive query rewriting, for each fact block type, there is a union of conjunctive queries that defines the certain answers query of the canonical query corresponding to the fact block type. Let k be the maximum number of variables occurring in these unions of conjunctive queries. We show that $\mathcal{M}$ is k-modular, and hence, by Theorem 3.9, $\mathcal{M}$ is definable by a finite set of s-t tgds.

Let $(I, J) \notin \mathcal{M}$. Let J^* be a universal solution of I whose fact blocks are of size at most n. Since J is a universal solution for I and $\mathcal{M}$ is closed under target homomorphisms, there is no homomorphism $h : J^* \to J$ constant on $adom(I)$. But then, it is not hard to see that there is a fact-block of J^* that cannot be homomorphically mapped into J by a homomorphism constant on I. Now, take the certain answer query corresponding to the fact-block type of this fact block. By assumption, it is definable by a union of conjunctive queries using at most k variables. But then, by the argument as in the proof of Theorem 3.2, there is a sub-instance $I' \subseteq I$ with $|adom(I')| \leq k$ such that J is already not a solution for I'. This shows that $\mathcal{M}$ is k-modular. □

3.4 Nested s-t tgds

We briefly consider an extension of s-t tgds, called *nested s-t tgds*. They generalize s-t tgds by allowing unlimited quantifier alternation. Nested s-t tgds are in fact the schema-mapping language that is used in the data exchange tool Clio, which has been developed at IBM Almaden [9].

DEFINITION 3.12. *Fix a partition of the set of first-order variables into two disjoint infinite sets, X and Y. A* nested s-t tgd *is a FO sentence that can be generated by the following recursive definition:*

$$\phi ::= \alpha \mid \forall x_1 \ldots x_n(\beta_1 \wedge \cdots \wedge \beta_k \rightarrow \exists y_1 \ldots y_m.(\phi_1 \wedge \cdots \wedge \phi_\ell))$$

where each $x_i \in X$, each $y_i \in Y$, α is any atomic formula over the target schema, and each β_i is an atomic formula over the source schema containing only variables from X, such that each x_i occurs in some β_j. Note that it is important that the above recursive definition may generate intermediate formulas with free variables, but the final result should be a sentence in order to qualify as a nested s-t tgd.

Nested s-t tgds extend the language of s-t tgds. In particular, the schema mapping from Example 3.7 can be defined by means of a nested s-t tgd, as follows: $\forall x_1 x_2 (Rx_1x_2 \rightarrow \exists y(Syx_2 \wedge \forall x_3(Rx_1x_3 \rightarrow Syx_3)))$.

PROPOSITION 3.13. *Every schema mapping defined by a finite set of nested s-t tgds is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting.*

PROOF. (Sketch) Closure under target homomorphisms can be shown by a straightforward formula induction. That nested s-t tgd mappings admit universal solutions and allow for conjunctive query rewriting follows from the fact that they can be translated into SO tgds [9], and the latter admit universal solutions [5] and allow for conjunctive query rewriting (Proposition 2.4). □

QUESTION 3.14. *Is it the case that a schema mapping is definable by a finite set of nested s-t tgds if and only if it is closed under target homomorphisms, admits universal solutions, and allows for conjunctive query rewriting?*

4. THE COMPLEXITY OF TESTING EXPRESSIBILITY

Our characterizations provide tools for testing whether a schema mapping specified in one language can also be defined in another language. In particular, it follows from our results that a schema mapping defined by a finite set of s-t tgds is definable by a finite

[1] The definition of *bounded fact block size* given in [6] was phrased in terms of *core universal solutions*. However, it is easy to see that the two are equivalent: if a source instance has any universal solution whose fact blocks have size at most n, then the *core* universal solution, being a sub-instance, will be such.

set of *full* s-t tgds if and only if it is closed under target intersection; and is definable by a finite set of *LAV* s-t tgds if and only if it is closed under union. In this section, we determine the *computational complexity* of testing definability in the different languages.

Testing whether a schema mapping specified by an FO sentence is expressible in the various schema-mapping languages we consider is undecidable. This follows from the undecidability of satisfiability for first-order sentences in the finite [15]: for any FO sentence ϕ over the source schema, consider the schema mapping defined by $\phi \rightarrow \neg\exists\mathbf{x}.R\mathbf{x}$ where R is a relation from the target schema. If ϕ is unsatisfiable, then the schema mapping is trivially definable in all schema-mapping languages we consider. If, on the other hand, ϕ is satisfiable, then the schema mapping is not definable in any of the languages, as it is not closed under target homomorphisms.

Hence, we will always assume that the input to the problem is a finite set of s-t tgds. The results are summarized in this table:

Input schema mapping	Desired schema mapping	Complexity of definability
s-t tgds	full s-t tgds	NP-complete
s-t tgds	LAV s-t tgds	NP-complete
full s-t tgds	LAV s-t tgds	PTIME
LAV s-t tgds	full s-t tgds	NP-complete

Our arguments are based on reductions from definability problems to the *entailment problem for s-t tgds*: given two schema mappings $\mathcal{M}_1, \mathcal{M}_2$, specified by a finite set of s-t tgds, is it the case that whenever $(I, J) \in \mathcal{M}_1$, also $(I, J) \in \mathcal{M}_2$? The complexity of the latter problem is established by the following lemma.

LEMMA 4.1. *The entailment problem for s-t tgds is NP-complete. It is in PTIME if the first schema mapping is specified by a finite set of LAV s-t tgds and the second is specified by a finite set of full s-t tgds.*

PROOF. The NP-hardness is by reduction from the containment problem for conjunctive queries: the conjunctive query $\exists\mathbf{y}'.\psi'$ is contained in the conjunctive query $\exists\mathbf{y}.\psi$ if and only if the s-t tgd $Px \rightarrow \exists\mathbf{y}.\psi$ entails the s-t tgd $Px \rightarrow \exists\mathbf{y}'.\psi'$. To see that the problem is *in* NP, let $\mathcal{M}_1, \mathcal{M}_2$ be schema mappings specified by a finite set of s-t tgds. In order to test whether $\mathcal{M}_1$ entails $\mathcal{M}_2$, we proceed as follows: for each s-t tgd

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

of $\mathcal{M}_2$, we use Proposition 2.5 to test in NP whether the certain answer query for $\exists\mathbf{y}.\psi(\mathbf{x}, \mathbf{y})$ with respect to $\mathcal{M}_1$ is satisfied in the canonical instance of $\phi(\mathbf{x})$, under the natural assignment for the variables $\mathbf{x}$.

For the second half of the result, let $\mathcal{M}_1, \mathcal{M}_2$ be two schema mappings, with $\mathcal{M}_1$ specified by a finite set of LAV s-t tgds and $\mathcal{M}_2$ specified by a finite set of full s-t tgds. In order to test whether $\mathcal{M}_1$ entails $\mathcal{M}_2$, we proceed as follows: for each s-t tgd $\forall\mathbf{x}(\phi \rightarrow \exists\mathbf{y}.\psi)$ of $\mathcal{M}_2$ and for each conjunct ϕ_i of ϕ, we take the canonical instance of ϕ_i and chase it with $\mathcal{M}_1$. In general, the chase may require exponential time, but in this case, since the source instance consists only of a single fact, it is not hard to see that it can be done in polynomial time. Finally, we test whether ψ holds in the union of the resulting instances (under the natural assignment) – again in PTIME. $\square$

We start by proving the upper bounds.

THEOREM 4.2. *Testing whether a schema mapping specified by a finite set of s-t tgds is definable by a finite set of full s-t tgds is in NP.*

PROOF. Let $\mathcal{M}$ by any schema mapping specified by a finite set of s-t tgds Σ. We first compute, in polynomial time, the "full part" Σ' of Σ by dropping all existential quantifiers and all conjuncts containing existentially quantified variables from the right-hand-sides of the dependencies. We will show that if $\mathcal{M}$ is defined by *any* finite set of full s-t tgds, then it is equivalent to the schema mapping $\mathcal{M}'$ defined by Σ'. It then follows by Lemma 4.1 that the problem is in NP.

Suppose $\mathcal{M}$ is definable by a finite set of s-t tgds, and in particular, is closed under target intersection. It is clear that $\mathcal{M}$ entails $\mathcal{M}'$, so it suffices to show that $\mathcal{M}'$ entails $\mathcal{M}$. Let I be any source instance and J any solution for I with respect to $\mathcal{M}'$. It follows from the construction of $\mathcal{M}'$ that there is a solution J' for I with respect to $\mathcal{M}$ such that $J \subseteq J'$ and J contains exactly those facts from J' that involve only elements from the domain of I (indeed, one may choose for J' the *canonical* universal solution of I with respect to $\mathcal{M}$). It follows by Proposition 2.8 that J is also a solution for I with respect to $\mathcal{M}$. This shows that $\mathcal{M}$ and $\mathcal{M}'$ are equivalent. $\square$

THEOREM 4.3. *Testing whether a schema mapping specified by a finite set of s-t tgds is definable by a finite set of LAV s-t tgds is in NP. It is in PTIME if the input consists of full s-t tgds.*

PROOF. Let $\mathcal{M}$ be any schema mapping specified by a finite set of s-t tgds Σ. We can define in a natural way the "LAV part" Σ' of Σ. The definition is in terms of *most general unifiers*. A *unifier* of a finite set of atomic formulas Φ is a variable substitution such that, after applying the substitution, all $\phi \in \Phi$ become identical. A *most general unifier* of Φ is a unifier σ of Φ such that for all other unifiers σ' of Φ, there is a variable substitution τ such that $\sigma'(x) = \tau(\sigma(x))$ for all variables x. It is well known that if a finite set of atomic formulas Φ has a unifier, then it has a most general unifier. Moreover, the existence of unifiers can be checked in polynomial time, and a most general unifier can be computed in polynomial time if it exists. We refer the reader to [2] for more information. Now, the "LAV part" Σ' of Σ is defined as follows: for each dependency, we test whether there is a unifier of the atomic formulas in the left-hand-side. If there is, then we compute a most general unifier, apply it to both sides of the dependency, and add the result to Σ'. If there is no such unifier (for instance because the left-hand-side contains atomic formulas using different relations), we simply ignore the dependency. It is clear from the construction that Σ' consists of LAV s-t tgds and that its size is polynomial in that of Σ. Let $\mathcal{M}'$ be the schema mapping defined by Σ. We will show that if $\mathcal{M}$ is definable by *any* finite set of LAV dependencies, then it is equivalent to $\mathcal{M}'$. It then follows by Lemma 4.1 that the problem is in NP, and in PTIME if the input is a full s-t tgd mapping.

Suppose $\mathcal{M}$ is definable by a finite set of LAV s-t tgds, and hence closed under union. It is clear that $\mathcal{M}$ entails $\mathcal{M}'$, so it suffices to show that $\mathcal{M}'$ entails $\mathcal{M}$. Let I, J be any pair of source and target instances such that J is a solution for I with respect to $\mathcal{M}'$. Let $I = \bigcup_i I_i$, where each I_i contains a single fact. Since $\mathcal{M}$ is closed under union, it suffices to show that J is a solution for each I_i with respect to $\mathcal{M}$. Suppose the antecedent of a dependency $\forall\mathbf{x}(\phi \rightarrow \exists\mathbf{y}.\psi)$ of $\mathcal{M}$ is satisfied in some I_i for some tuple $\mathbf{d}$, i.e., $I_i \models \phi(\mathbf{d})$. Since I_i contains only a single fact, this implies that there is a unifier for all conjuncts of ϕ, namely the one corresponding to the kernel of the map sending $\mathbf{x}$ to $\mathbf{d}$. Hence, the conjuncts of ϕ also have a *most general* unifier, and the LAV s-t tgd of $\mathcal{M}'$ that was obtained from the above s-t tgd by this most general unifier ensures that the right hand side $\exists\mathbf{y}.\psi$ is satisfied (for the tuple $\mathbf{d}$). Thus, J is a solution for each I_i, and hence, by closure under union, for

I. $\square$

We now proceed with the lower bounds.

THEOREM 4.4. *Deciding whether a schema mapping specified by a finite set of s-t tgds is definable by a finite set of LAV s-t tgds is NP-hard.*

PROOF. We give a reduction from the Boolean conjunctive query containment problem. Let two conjunctive queries $\exists \mathbf{x}.\phi$ and $\exists \mathbf{y}.\psi$ be given, and let P, Q be unary relation symbols occurring in neither of the two queries. Then the schema mapping consisting of the s-t tgds $Pz \rightarrow \exists \mathbf{x}.\psi$ and $Pz \wedge Qz \rightarrow \exists \mathbf{y}.\psi'$ is closed under union if and only if $\exists \mathbf{y}.\psi'$ is contained in $\exists \mathbf{x}.\phi$. Indeed, if the containment holds, then the second s-t tgd above is redundant, and hence the schema mapping can be defined using only LAV s-t tgds. If, on the other hand, the containment does not hold, then it is easily seen that the given schema mapping fails to be closed under union on the source instance containing only the two facts Pa, Qa (for some value a). In particular, the schema mapping is not definable by LAV s-t tgds. $\square$

THEOREM 4.5. *Testing whether a schema mapping specified by a finite set of LAV s-t tgds is definable by a finite set of full s-t tgds is NP-hard.*

PROOF. We again give a reduction from the Boolean conjunctive query containment problem. Let conjunctive queries $\exists \mathbf{x}.\phi$ and $\exists \mathbf{y}.\psi$ be given. Let R be a fresh relation symbol whose arity is equal to the number of variables in the sequence $\mathbf{x}$, and consider the schema mapping $\mathcal{M}$ defined by the s-t tgds $R\mathbf{x} \rightarrow \phi$ and $R\mathbf{x} \rightarrow \exists \mathbf{y}.\psi$. Observe that both s-t tgds are LAV and only the second one is not full.

Now, $\mathcal{M}$ is definable by a finite set of full s-t tgds if and only if $\exists \mathbf{y}.\psi$ is contained in $\exists \mathbf{x}.\phi$. Indeed, if the containment holds, the second s-t tgd of $\mathcal{M}$ is redundant, and hence $\mathcal{M}$ definable by a finite set of full s-t tgds. If, on the other hand, the containment does not hold, then let J_1 be a witnessing instance. Then there is a tuple $\mathbf{a}$ such that $J_1 \models \phi(\mathbf{a})$ but $J_1 \not\models \exists \mathbf{y}.\psi$. Let J_2 and J_3 be two isomorphic copies of the canonical instance of $\exists \mathbf{y}.\psi$. We may assume that J_1, J_2, J_3 have disjoint domains. Finally, let $I = \{R\mathbf{a}\}$. By construction, both $J_1 \cup J_2$ and $J_1 \cup J_3$ are solutions for I with respect to $\mathcal{M}$, but their intersection, being J_1, is not. Hence, $\mathcal{M}$ is not closed under target intersections, and therefore not definable by full s-t tgds. $\square$

5. INSTANCE-LEVEL DEFINABILITY

In this section we briefly consider a different kind of expressive power: we ask which target instances can be obtained from source instances by means of schema mappings, in the spirit of the concept of BP-completeness from [3, 13]. For schema mappings definable by a finite set of s-t tgds, the answer is given by the following theorem:

THEOREM 5.1. *For every source instance I and every target instance J, the following are equivalent:*

1. *J is a universal solution for I with respect to some schema mapping $\mathcal{M}$ definable by a finite set of s-t tgds*
2. *Every homomorphism $h : I \rightarrow I$ extends to a homomorphism $\widehat{h} : J \rightarrow J$*

PROOF. The direction (1) $\Rightarrow$ (2) follows from the fact that schema mappings definable by finite sets of s-t tgds reflect source homomorphisms (cf. Proposition 2.9). For the other direction, assume that every homomorphism $h : I \rightarrow I$ extends to a homomorphism $\widehat{h} : J \rightarrow J$. Fix variables $x_1, x_2, \ldots$ for the elements of I and variables $y_1, y_2, \ldots$ for the elements of J that are not elements of I. Let $\Sigma(\mathbf{x})$ be set of all facts true in I and $\Theta(\mathbf{x}, \mathbf{y})$ the set of all facts true in J. Let $\mathcal{M}$ be the schema mapping with the single s-t tgd $\forall \mathbf{x}(\bigwedge \Sigma(\mathbf{x}) \rightarrow \exists \mathbf{y}(\bigwedge \Theta(\mathbf{x}, \mathbf{y}))$. We claim that J is a universal solution for I with respect to $\mathcal{M}$.

To see that J is a solution, let g be any assignment under which $\bigwedge \Sigma(\mathbf{x})$ is satisfied in I. Then g is in effect a homomorphism from I to I. Hence, it extends to a homomorphism $\widehat{g} : J \rightarrow J$. Hence, $\bigwedge \Theta(\mathbf{x}, \mathbf{y})$ is true in J under the assignment $\widehat{g}$, and therefore, $\exists \mathbf{y}.(\bigwedge \Theta(\mathbf{x}, \mathbf{y})$ is true in J under the assignment g. This shows that J is a solution of I with respect to $\mathcal{M}$. To see that J is a *universal* solution, let J' be any other solution of I with respect to $\mathcal{M}$. Then J' must satisfy $\exists \mathbf{y} \bigwedge \Theta$ under the natural assignment. This just shows that there is a homomorphism from J to J' constant on $adom(I)$. $\square$

Incidentally, it follows that J is a *core* universal solution for I with respect to some schema mapping definable by finitely many s-t tgds if and only if J is a core and every homomorphism $h : I \rightarrow I$ extends to a homomorphism $\widehat{h} : J \rightarrow J$.

It may also be worth mentioning that another characterization along the same lines can be obtained for dependencies with inequalities. Define *source-to-target dependency with inequalities* like s-t tgds except that the left-hand-side of the implication may contain inequalities. Then a target instance J is a universal solution for a source instance I with respect to some schema mapping definable a finite set of such dependencies if and only if every automorphism $h : I \rightarrow I$ extends to a homomorphism $\widehat{h} : J \rightarrow J$. The proof is similar to that of Theorem 5.1, the only difference being that the set of atomic formulas Σ now also contains inequalities $x_i \neq x_j$ for $i \neq j$. Any assignment making Σ true in I is then in fact an automorphism (it is a bijective homomorphism, and the total number of true facts cannot increase). Given that universal solutions are only unique up to homomorphic equivalence, this shows that source-to-target dependencies with inequalities are, in a natural sense, "BP-complete" for data exchange purposes. The inspiration for this observation comes from a remark by L. Libkin (personal communication), who pointed out to us that source-to-target dependencies with arbitrary first-order left-hand-sides are BP-complete in the same sense.

6. CONCLUDING REMARKS

In this paper, we have focused on structural characterizations of schema mappings specified by s-t tgds. There are, however, other important schema-mapping languages of source-to-target dependencies for which this type of investigation remains to be carried out. The next step would be to pursue structural characterizations of schema mappings specified by finite sets of nested s-t tgds and by SO tgds. Furthermore, one may also consider schema-mapping languages that include target constraints, such as target tuple-generating dependencies and target equality-generating dependencies. In this context, it is worth mentioning the work of Makowsky and Vardi [12], which provides characterizations for various classes of data dependencies; however, these characterizations either concern arbitrary (finite and infinite) structures or they are about definability by infinite sets of dependencies. An interesting question is whether there is a natural way to characterize weakly acyclic sets of target tgds [5], a class of target dependencies that is of central importance in data exchange, as they guarantee termination of the chase procedure within a polynomial number of steps.

In a different direction, we note that recent work on schema-mapping optimization [6] has considered *data-exchange equiva-*

lence and *conjunctive-query equivalence*, two notions of equivalence between schema mappings that are strictly weaker than logical equivalence, but sufficient for data-exchange purposes or for query-answering purposes. The work in [6] has addressed mainly the question of when a schema mapping specified in a richer language (say, SO-tgds) is conjunctive-query equivalent to a schema mapping specified in a simpler language (say, s-t tgds). We leave it as an open problem to obtain structural characterizations of schema mappings in the context of conjunctive-query equivalence or in data-exchange equivalence.

7. REFERENCES

[1] S. Abiteboul and O. M. Duschka. Complexity of Answering Queries Using Materialized Views. In *ACM Symposium on Principles of Database Systems (PODS)*, pages 254–263, 1998.

[2] F. Baader and J. H. Siekmann. Unification theory. In *Handbook of logic in artificial intelligence and logic programming*, pages 41–125. Oxford University Press, Inc., New York, NY, USA, 1994.

[3] F. Bancilhon. On the completeness of query languages for relational data bases. In *MFCS*, pages 112–123, 1978.

[4] O. M. Duschka and M. R. Genesereth. Answering recursive queries using views. In *ACM Symposium on Principles of Database Systems (PODS)*, pages 109–116, 1997.

[5] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005. Preliminary version in ICDT 2003.

[6] R. Fagin, P. G. Kolaitis, A. Nash, and L. Popa. Towards a theory of schema-mapping optimization. In M. Lenzerini and D. Lembo, editors, *PODS*, pages 33–42. ACM, 2008.

[7] R. Fagin, P. G. Kolaitis, L. Popa, and W.-C. Tan. Composing Schema Mappings: Second-order Dependencies to the Rescue. *ACM Transactions on Database Systems (TODS)*, 30(4):994–1055, 2005. A preliminary version of this paper appeared in the 2004 PODS conference.

[8] R. Fagin, P. G. Kolaitis, L. Popa, and W. C. Tan. Quasi-inverses of schema mappings. *ACM Trans. Database Syst.*, 33(2), 2008. Preliminary version in PODS 2007.

[9] A. Fuxman, M. A. Hernandez, H. Ho, R. J. Miller, P. Papotti, and L. Popa. Nested mappings: schema mapping reloaded. In *VLDB*, pages 67–78, 2006.

[10] P. G. Kolaitis. Schema Mappings, Data Exchange, and Metadata Management. In *ACM Symposium on Principles of Database Systems (PODS)*, pages 61–75, 2005.

[11] M. Lenzerini. Data Integration: A Theoretical Perspective. In *ACM Symposium on Principles of Database Systems (PODS)*, pages 233–246, 2002.

[12] J. A. Makowsky and M. Y. Vardi. On the expressive power of data dependencies. *Acta Informatica*, 23(3):231–244, 1986.

[13] J. Paredaens. On the expressive power of the relational algebra. *Inf. Process. Lett.*, 7(2):107–111, 1978.

[14] B. Rossman. Existential positive types and preservation under homomorphisisms. In *Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science (LICS'05)*, pages 467–476, Washington, DC, USA, 2005. IEEE Computer Society.

[15] B. Trakhtenbrot. Impossibility of an algorithm for the decision problem on finite classes. *Dokl. Akad. Nauk. SSSR*, 70:569–572, 1950.