Composing Schema Mappings: Second-Order Dependencies to the Rescue

RONALD FAGIN, PHOKION G. KOLAITIS, and LUCIAN POPA
IBM Almaden Research Center
and
WANG-CHIEW TAN
University of California, Santa Cruz

A schema mapping is a specification that describes how data structured under one schema (the source schema) is to be transformed into data structured under a different schema (the target schema). A fundamental problem is composing schema mappings: given two successive schema mappings, derive a schema mapping between the source schema of the first and the target schema of the second that has the same effect as applying successively the two schema mappings.

In this article, we give a rigorous semantics to the composition of schema mappings and investigate the definability and computational complexity of the composition of two schema mappings. We first study the important case of schema mappings in which the specification is given by a finite set of source-to-target tuple-generating dependencies (source-to-target tgds). We show that the composition of a finite set of full source-to-target tgds with a finite set of tgds is always definable by a finite set of source-to-target tgds, but the composition of a finite set of source-to-target tgds with a finite set of full source-to-target tgds may not be definable by any set (finite or infinite) of source-to-target tgds; furthermore, it may not be definable by any formula of least fixed-point logic, and the associated composition query may be NP-complete. After this, we introduce a class of existential second-order formulas with function symbols and equalities, which we call *second-order tgds*, and make a case that they are the "right" language for composing schema mappings. Specifically, we show that second-order tgds form the smallest class (up to logical equivalence) that contains every source-to-target tgd and is closed under conjunction and composition. Allowing equalities in second-order tgds turns out to be of the essence, even though the "obvious" way to define second-order tgds does not require equalities. We show that second-order tgds without equalities are not sufficiently expressive to define the composition of finite sets of source-to-target tgds. Finally, we show that second-order tgds possess good properties for data exchange and query answering: the

P. G. Kolaitis did this research while he was on leave from UC Santa Cruz.
W.-C. Tan was supported in part by National Science Foundation (NSF) CAREER Award IIS-0347065 and NSF grant IIS-0430994.
A preliminary version of this article appeared in *Proceedings of the 2004 ACM Symposium of Principles of Database Systems* (Paris, France), ACM, New York, 2004, pp. 83–94.

Authors' addresses: R. Fagin, P. G. Kolaitis, and L. Popa, IBM Almaden Research Center, 650 Harry Road, San Jose, CA 95120; W.-C. Tan, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA 95064. Correspondence email: fagin@almaden.ibm.com.

chase procedure can be extended to second-order tgds so that it produces polynomial-time computable universal solutions in data exchange settings specified by second-order tgds.

Categories and Subject Descriptors: H.2.4 [**Database Management**]: Systems—*Relational databases; query processing*; H.2.5 [**Database Management**]: Heterogeneous Databases—*Data translation*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Data exchange, data integration, composition, schema mapping, certain answers, conjunctive queries, dependencies, chase, computational complexity, query answering, second-order logic, universal solution, metadata model management

1. INTRODUCTION & SUMMARY OF RESULTS

The problem of transforming data structured under one schema into data structured under a different schema is an old, but persistent problem, arising in several different areas of database management systems. In recent years, this problem has received considerable attention in the context of information integration, where data from various heterogeneous sources has to be transformed into data structured under a mediated schema. To achieve interoperability, data-sharing architectures use *schema mappings* to describe how data is to be transformed from one representation to another. These schema mappings are typically specified using high-level declarative formalisms that make it possible to describe the correspondence between different schemas at a logical level, without having to specify physical details that may be relevant only for the implementation (run-time) phase. In particular, declarative schema mappings in the form of GAV (global-as-view), LAV (local-as-view), and, more generally, GLAV (global-and-local-as-view) assertions have been used in *data integration* systems [Lenzerini 2002]. Similarly, source-to-target tuple-generating dependencies (source-to-target tgds) have been used for specifying data exchange between a relational source and a relational target [Fagin et al. 2005a, 2005b]; moreover, nested (XML-style) source-to-target dependencies have been used in the Clio data exchange system [Popa et al. 2002].

The extensive use of schema mappings has motivated the need to develop a framework for managing these schema mappings and other related metadata. Bernstein [2003] has introduced such a framework, called *model management*, in which the main abstractions are schemas and mappings between schemas, as well as operators for manipulating schemas and mappings. One of the most fundamental operators in this framework is the *composition operator*, which combines successive schema mappings into a single schema mapping. The composition operator can play a useful role each time the target of a schema mapping is also the source of another schema mapping. This scenario occurs, for instance, in schema evolution, where a schema may undergo several successive changes. It also occurs in peer-to-peer data management systems, such as the Piazza System [Halevy et al. 2003], and in extract-transform-load (ETL) processes in which the output of a transformation may be input to another [Vassiliadis et al. 2002]. A model management system should be able to figure out automatically how to compose two or more successive schema mappings into a single schema mapping between the first schema and the last schema in a way that captures the interaction of the schema mappings in the entire sequence. The resulting single schema mapping can then be used during the run-time phase for various purposes, such as query answering and data exchange, potentially with significant performance benefits.

Bernstein's approach provides a rich conceptual framework for model management. The next stage in the development of this framework is to provide a rigorous and meaningful semantics of the basic model management operators and to investigate the properties of this semantics. As pointed out by Bernstein [2003], while the semantics of the match operator have been worked out to a certain extent, the semantics of other basic operators, including the composition operator, "are less well developed". The problem of composing schema mappings has the following general formulation: given a schema mapping $\mathcal{M}_{12}$ from schema $\mathbf{S}_1$ to schema $\mathbf{S}_2$, and a schema mapping $\mathcal{M}_{23}$ from schema $\mathbf{S}_2$ to schema $\mathbf{S}_3$, derive a schema mapping $\mathcal{M}_{13}$ from schema $\mathbf{S}_1$ to schema $\mathbf{S}_3$ that is "equivalent" to the successive application of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$. Thus, providing semantics to the composition operator amounts to making precise what "equivalence" means in this context. Madhavan and Halevy [2003] were the first to propose a semantics of the composition operator. To this effect, they defined the semantics of the composition operator relative to a class $\mathcal{Q}$ of queries over the schema $\mathbf{S}_3$ by stipulating that "equivalence" means that, for every query q in $\mathcal{Q}$, the certain answers of q in $\mathcal{M}_{13}$ coincide with the certain answers of q that would be obtained by successively applying the two schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$. They then established a number of results for this semantics in the case in which the schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ are specified by source-to-target tgds (i.e., GLAV assertions), and the class $\mathcal{Q}$ is the class of all conjunctive queries over $\mathbf{S}_3$. The semantics of the composition operator proposed by Madhavan and Halevy [2003] is a significant first step, but it suffers from certain drawbacks that seem to be caused by the fact that this semantics is given relative to a class of queries. To begin with, the set of formulas specifying a composition $\mathcal{M}_{13}$ of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ relative to a class $\mathcal{Q}$ of queries need not be unique up to logical equivalence, even when the class $\mathcal{Q}$ of queries is held fixed. Moreover, this semantics is rather fragile, because as we show, a schema mapping $\mathcal{M}_{13}$ may be a composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ when $\mathcal{Q}$ is the class of conjunctive queries (the class $\mathcal{Q}$ that Madhavan and Halevy focused on), but fail to be a composition of these two schema mappings when $\mathcal{Q}$ is the class of conjunctive queries with inequalities.

In this article, we first introduce a different semantics for the composition operator and then investigate the definability and computational complexity of the composition of schema mappings under this new semantics. Unlike the semantics proposed by Madhavan and Halevy, our semantics does not carry along a class of queries as a parameter. Specifically, we focus on the space of instances of schema mappings and define a schema mapping $\mathcal{M}_{13}$ to be a composition of two schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ if the space of instances of $\mathcal{M}_{13}$ is the set-theoretic composition of the spaces of instances of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, where these spaces are viewed as binary relations between source instances and target instances. One advantage of this approach is that the set of formulas defining a composition $\mathcal{M}_{13}$ of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is unique up to logical equivalence; thus, we can refer to such a schema mapping $\mathcal{M}_{13}$ as *the* composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$. Moreover, our semantics is robust, since it is defined in terms of the schema mappings alone and without reference to a set of queries. In fact, it is easy to see that the composition (in our sense) of two schema mappings is a composition of these two schema mappings in the sense of Madhavan and Halevy relative to *every* class of queries.

We explore in depth the properties of the composition of schema mappings specified by a finite set of source-to-target tuple-generating dependencies (source-to-target tgds). A natural question to ask is whether the composition of two such schema mappings can also be specified by a finite set of source-to-target tgds; if not, in what logical formalism can it be actually expressed? On the positive side, we show that the composition of a finite set of full source-to-target tgds with a finite set of source-to-target tgds is always definable by a finite set of source-to-target tgds (a source-to-target tgd is *full* if no existentially quantified variables occur in the tgd). On the negative side, however, we show that the composition of a finite set of source-to-target tgds with a finite set of full source-to-target tgds may not be definable by any set (finite or infinite) of source-to-target tgds. We also show that the composition of a finite set of source-to-target tgds with a finite set of full source-to-target tgds may not even be definable in the finite-variable infinitary logic $L^{\omega}_{\infty\omega}$, which implies that it is not definable in least fixed-point logic LFP; moreover, the associated composition query can be NP-complete.

To ameliorate these negative results, we introduce a class of existential second-order formulas with function symbols and equalities, called *second-order tgds*, which express source-to-target constraints and which subsume the class of finite conjunctions of (first-order) source-to-target tgds. We make a case that second-order tgds are the *right* language both for specifying schema mappings and for composing such schema mappings. To begin with, we show that the composition of two finite sets of source-to-target tgds is always definable by a second-order tgd. Moreover, the composition of second-order tgds is also definable by a second-order tgd, and we give an algorithm that, given two schema mappings specified by second-order tgds, outputs a second-order tgd that defines the composition. Furthermore, the conjunction of a finite set of second-order tgds is equivalent to a single second-order tgd. Hence, the composition of a finite number of schema mappings, each defined by a finite set of source-to-target (second-order) tgds, is always definable by a second-order tgd. It should be pointed out that arriving at the *right* concept of second-order tgds is a rather delicate matter. Indeed, at first one may consider the class of second-order formulas that are obtained from first-order tgds by Skolemizing the existential first-order quantifiers into existentially quantified function symbols. This process gives rise to a class of existential second-order formulas with no equalities. Therefore, the “obvious” way to define second-order tgds is with formulas with no equalities. Interestingly enough, however, we show that second-order tgds without equalities are *not* sufficiently expressive to define the composition of finite sets of (first-order) source-to-target tgds. In fact, our second-order tgds (with equalities) form the smallest class of formulas (up to logical equivalence) for composing schema mappings given by finite sets of source-to-target tgds; every second-order tgd defines the composition of a finite sequence of schema mappings, each defined by a finite set of source-to-target tgds.

We then show that second-order tgds possess good properties for data exchange. In particular, the chase procedure can be extended to second-order tgds so that it produces polynomial-time computable "universal solutions" (in the sense of Fagin et al. [2005a]) in data exchange settings specified by second-order tgds. As a result, in such data exchange settings the certain answers of conjunctive queries can be computed in polynomial time.

In spite of the richness of second-order tgds, they form a well-behaved fragment of second-order logic for composing schema mappings. As we noted earlier, if the data exchange setting is defined by second-order tgds, then the certain answers of every conjunctive query can be computed in polynomial time (by doing the chase). By contrast, when the source schema is described in terms of the target schema by means of arbitrary first-order views, there are conjunctive queries for which computing the certain answers is an undecidable problem [Abiteboul and Duschka 1998]. Thus, our second-order tgds form a fragment of second-order logic that in some ways is more well-behaved than first-order logic.

There is a subtle issue about the choice of universe in the semantics of second-order tgds. We take our universe to be a countably infinite set of elements that includes the active domain. This is a natural choice for the universe, since second-order tgds have existentially quantified function symbols and for this reason, one needs sufficiently many elements in the universe in order to interpret these function symbols without making any unnecessary combinatorial assumptions. In fact, we show that as long as we take the universe to be finite but sufficiently large, then the semantics of a second-order tgd remains unchanged from the infinite universe semantics.

We show that determining whether a given instance over the source and target schema satisfies a second-order tgd is in NP and can be NP-complete. This is in contrast with the first-order case, where such "model checking" can be done in polynomial time.

Finally, we examine Madhavan and Halevy's notion of composition, which we refer to as "certain-answer adequacy". Roughly speaking, a formula is certain-answer adequate if it gives the same certain answers as the composition. A formula σ that defines the composition (in our sense) is always certain-answer adequate for every class $\mathcal{Q}$ of queries; however, other formulas that are not logically equivalent to σ may also be certain-answer adequate for some classes $\mathcal{Q}$ of queries. This is why we use the word "adequate": logically inequivalent choices may both be adequate for the job. We show that there are schema mappings where no finite set of source-to-target tgds is certain-answer adequate for conjunctive queries. We also prove the following "hierarchy" of results about certain-answer adequacy:

(A) A formula may be certain-answer adequate for conjunctive queries but not for conjunctive queries with inequalities.

(B) A formula may be certain-answer adequate for conjunctive queries with inequalities but not for all first-order queries.

(C) A formula is certain-answer adequate for all first-order queries if and only if it defines the composition (in our sense); furthermore, such a formula is certain-answer adequate for all queries. It follows that if a formula is certain-answer adequate for all first-order queries, then it is certain-answer adequate for all queries.

2. BACKGROUND

In this section, we review the basic concepts from data exchange that we will need.

A *schema* is a finite sequence $\mathbf{R} = \langle R_1, \ldots, R_k \rangle$ of distinct relation symbols, each of a fixed arity. An *instance* I (*over the schema* $\mathbf{R}$) is a sequence $\langle R_1^I, \ldots, R_k^I \rangle$ such that each R_i^I is a finite relation of the same arity as R_i. We call R_i^I the R_i*-relation of* I. We shall often abuse the notation and use R_i to denote both the relation symbol and the relation R_i^I that interprets it.

Let $\mathbf{S} = \langle S_1, \ldots, S_n \rangle$ and $\mathbf{T} = \langle T_1, \ldots, T_m \rangle$ be two schemas with no relation symbols in common. We write $\langle \mathbf{S}, \mathbf{T} \rangle$ to denote the schema $\langle S_1, \ldots, S_n, T_1, \ldots, T_m \rangle$. If I is an instance over $\mathbf{S}$ and J is an instance over $\mathbf{T}$, then we write $\langle I, J \rangle$ for the instance K over the schema $\langle \mathbf{S}, \mathbf{T} \rangle$ such that $S_i^K = S_i^I$ and $T_j^K = T_j^J$, for $1 \leq i \leq n$ and $1 \leq j \leq m$.

If K is an instance and σ is a formula in some logical formalism, then we write $K \models \sigma$ to mean that K satisfies σ. If Σ is a set of formulas, then we write $K \models \Sigma$ to mean that $K \models \sigma$ for every formula $\sigma \in \Sigma$. Recall that a *(Boolean) query* is a class of instances that is closed under isomorphisms [Chandra and Harel 1982]. That is, if a structure is a member of the class, then so is every isomorphic copy of the structure. If K is an instance and q is a query, then we write $K \models q$ to mean that K is a member of the class q of instances.

Definition 2.1. A *schema mapping* (or, in short, *mapping*) is a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ and $\mathbf{T}$ are schemas with no relation symbols in common and Σ is a set of formulas of some logical formalism over $\langle \mathbf{S}, \mathbf{T} \rangle$.

Definition 2.2. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping.

(1) An *instance of* $\mathcal{M}$ is an instance $\langle I, J \rangle$ over $\langle \mathbf{S}, \mathbf{T} \rangle$ that satisfies every formula in the set Σ.

(2) We write Inst($\mathcal{M}$) to denote the set of all instances $\langle I, J \rangle$ of $\mathcal{M}$. Moreover, if $\langle I, J \rangle \in$ Inst($\mathcal{M}$), then we say that J is a *solution for* I *under* $\mathcal{M}$.

Several remarks are in order now. In the sequel, if $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a schema mapping, we will often refer to $\mathbf{S}$ as the *source schema* and to $\mathbf{T}$ as the *target schema*. The formulas in the set Σ express *constraints* that an instance $\langle I, J \rangle$ over the schema $\langle \mathbf{S}, \mathbf{T} \rangle$ must satisfy. We assume that the logical formalisms considered have the property that the satisfaction relation between formulas and instances is *preserved under isomorphism*, which means that if an instance satisfies a formula, then every isomorphic instance also satisfies that formula. This is a mild condition that is true of all standard logical formalisms, such as first-order logic, second-order logic, fixed-point logics, and infinitary logics. Thus, such formulas represent queries in the sense of Chandra and Harel [1982]. An immediate consequence of this property is that Inst($\mathcal{M}$) is *closed under isomorphism*; that is, if $\langle I, J\rangle \in$ Inst($\mathcal{M}$) and $\langle I', J'\rangle$ is isomorphic to $\langle I, J\rangle$, then also $\langle I', J'\rangle \in$ Inst($\mathcal{M}$).

At this level of generality, some of the formulas in Σ may be just over the source schema $\mathbf{S}$ and others may be just over the target schema $\mathbf{T}$; thus, the set Σ may include constraints over the source schema $\mathbf{S}$ alone or over the target schema $\mathbf{T}$ alone, along with constraints that involve both the source and the target schemas. We note that, although the term "schema mapping" or "mapping" has been used earlier in the literature (for instance, in Miller et al. [2000], Madhavan and Halevy [2003]), it is a bit of a misnomer, as a schema mapping is not a mapping in the traditional mathematical sense, but actually it is a schema (although partitioned in two parts) together with a set of constraints. Nonetheless, a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ gives rise to a mapping such that, given an instance I over $\mathbf{S}$, it associates the set of all instances J over $\mathbf{T}$ that are solutions for I under $\mathcal{M}$. Note also that the terminology "J is a solution for I" comes from Fagin et al. [2005a, 2005b], where J is a solution to the *data exchange problem* associated with the mapping $\mathcal{M}$ and the source instance I.

Schema mappings are often specified using *source-to-target tgds*. They have been used to formalize data exchange [Fagin et al. 2005a, 2005b]. They have also been used in data integration scenarios under the name of GLAV assertions [Lenzerini 2002]. A *source-to-target tuple-generating dependency (source-to-target tgd)* is a first-order formula of the form

$$\forall \mathbf{x}(\phi_S(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi_T(\mathbf{x}, \mathbf{y})),$$

where $\phi_S(\mathbf{x})$ is a conjunction of atomic formulas over $\mathbf{S}$, and where $\psi_T(\mathbf{x}, \mathbf{y})$ is a conjunction of atomic formulas over $\mathbf{T}$. We assume that every variable in $\mathbf{x}$ appears in ϕ_S. A *full source-to-target tuple-generating dependency (full source-to-target tgd)* is a source-to-target tgd of the form

$$\forall \mathbf{x}(\phi_S(\mathbf{x}) \rightarrow \psi_T(\mathbf{x})),$$

where $\phi_S(\mathbf{x})$ is a conjunction of atomic formulas over $\mathbf{S}$, and where $\psi_T(\mathbf{x})$ is a conjunction of atomic formulas over $\mathbf{T}$. We again assume that every variable in $\mathbf{x}$ occurs in ϕ_S.

Every full source-to-target tgd is logically equivalent to a finite set of full source-to-target tgds each of which has a single atom in its right-hand side. Specifically, a full source-to-target tgd of the form $\forall \mathbf{x}(\phi_S(\mathbf{x}) \rightarrow \wedge_{i=1}^{k} R_i(\mathbf{x}_i))$ is equivalent to the set consisting of the full source-to-target tgds $\forall \mathbf{x}(\phi_S(\mathbf{x}) \rightarrow R_i(\mathbf{x}_i))$, for $i = 1, \ldots, k$. In contrast, this property fails for arbitrary source-to-target tgds, since the existential quantifiers may bind variables used across different atomic formulas.

Example 2.3. Consider the following three schemas $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$. Schema $\mathbf{S}_1$ consists of a single binary relation symbol `Takes`, that associates student names with the courses they take. Schema $\mathbf{S}_2$ consists of a similar binary relation symbol `Takes`$_1$, that is intended to provide a copy of `Takes`, and of an additional binary relation symbol `Student`, that associates each student name with a student id. Schema $\mathbf{S}_3$ consists of one binary relation symbol `Enrollment`, that associates student ids with the courses the students take. Consider now the schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where

$$\begin{aligned}
\Sigma_{12} &= \{\forall n \forall c(\texttt{Takes}(n, c) \rightarrow \texttt{Takes}_1(n, c)),\\
&\quad\;\; \forall n \forall c(\texttt{Takes}(n, c) \rightarrow \exists s \texttt{Student}(n, s))\}\\
\Sigma_{23} &= \{\forall n \forall s \forall c(\texttt{Student}(n, s) \wedge \texttt{Takes}_1(n, c) \rightarrow \texttt{Enrollment}(s, c))\}
\end{aligned}$$

The three formulas in Σ_{12} and Σ_{23} are source-to-target tgds. The second formula in Σ_{12} is an example of a source-to-target tgd that is not full, while the other two formulas are full source-to-target tgds. The first mapping, associated with the set Σ_{12} of formulas, requires that "copies" of the tuples in Takes must exist in $\texttt{Takes}_1$ and, moreover, that each student name n must be associated with some student id s in Student. The second mapping, associated with the formula in Σ_{23}, requires that pairs of student id and course must exist in the relation Enrollment, provided that they are associated with the same student name.

Note that for a given set Σ of source-to-target tgds, checking whether an instance $\langle I, J \rangle$ satisfies Σ can be done in polynomial time. (This is true in general when Σ is a set of first-order formulas.) We shall contrast this with the case of *second-order tgds*, the more expressive mapping language that we shall introduce later. When Σ is a second-order tgd, checking if $\langle I, J \rangle$ satisfies Σ is in NP and can be NP-complete (Theorem 5.7).

For the rest of this section, we shall review notions and results from Fagin et al. [2005a] about data exchange. The *data exchange problem associated with $\mathcal{M}$ and a source instance I* is to find a solution J over the target schema $\mathbf{T}$. For any schema mapping $\mathcal{M}$, there may be many solutions for a given source instance I over $\mathbf{S}$. Let $\mathbf{R}$ be a schema and J, J' two instances over $\mathbf{R}$. A function h is a *homomorphism* from J to J' if for every relation symbol R in $\mathbf{R}$ and every tuple $(a_1, \ldots, a_n) \in R^J$, we have that $(h(a_1), \ldots, h(a_n)) \in R^{J'}$. Given a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and a source instance I over S, a *universal solution for I* under $\mathcal{M}$ is a solution J of I under $\mathcal{M}$ such that for every solution J' of I under $\mathcal{M}$, there exists a homomorphism $h : J \rightarrow J'$ with the property that $h(v) = v$ for every value v that occurs in I. Intuitively, universal solutions are the "best" solutions among the space of all solutions for I. If Σ consists of source-to-target tgds, then *chasing* I with Σ produces a universal solution J of I under $\mathcal{M}$. Furthermore, J can be computed in time polynomial in the size of I. (This holds even in a more general setting that also includes target constraints.) We will refer to this result several times during the technical development of this paper. During the chase, target values may be introduced that do not appear in the source instance; these are called *nulls*.

Given a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, an instance I over the source schema $\mathbf{S}$ and a k-ary query q posed against the target schema $\mathbf{T}$, the *certain answers of q on I with respect to $\mathcal{M}$*, denoted by $certain_{\mathcal{M}}(q, I)$, is the set of all k-tuples t of values from I such that, for every solution J of I under $\mathcal{M}$, we have that $t \in q(J)$, where $q(J)$ is the result of evaluating q on J. If J is a universal solution for I under $\mathcal{M}$, and q is a union of conjunctive queries, then $certain_{\mathcal{M}}(q, I)$ equals $q(J)_{\downarrow}$, which is the result of evaluating q on J and then keeping only those tuples formed entirely of values from I (i.e., tuples that do not contain nulls). The equality $certain_{\mathcal{M}}(q, I) = q(J)_{\downarrow}$ holds for arbitrarily specified schema mappings $\mathcal{M}$ (as long as such a universal solution J exists). Since a universal solution can be computed in time polynomial in the size of I for schema mappings that contain only source-to-target tgds, it follows that the certain answers of q on I with respect to such schema mappings can also be computed in polynomial time.

3. THE SEMANTICS OF COMPOSITION

In this section, we define what it means for a schema mapping to be the composition of two schema mappings. In later sections, we will investigate under what conditions such schema mappings exist and in what language they can be defined.

If P_1 and P_2 are two binary relations, then by definition, the *composition* $P_1 \circ P_2$ of P_1 and P_2 is the binary relation

$$P_1 \circ P_2 = \{(x, y) : (\exists z)((x, z) \in P_1 \wedge (z, y) \in P_2)\}.$$

Clearly, if $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a schema mapping, then $\text{Inst}(\mathcal{M})$ is a binary relation between instances over $\mathbf{S}$ and instances over $\mathbf{T}$. In what follows, we define the concept of a *composition of two schema mappings* $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ using the composition of the binary relations $\text{Inst}(\mathcal{M}_{12})$ and $\text{Inst}(\mathcal{M}_{23})$.

Definition 3.1. Let $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ be two schema mappings such that the schemas $\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3$ have no relation symbol in common pairwise. A schema mapping $\mathcal{M} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ is a *composition of* $\mathcal{M}_{12}$ *and* $\mathcal{M}_{23}$ if

$$\text{Inst}(\mathcal{M}) = \text{Inst}(\mathcal{M}_{12}) \circ \text{Inst}(\mathcal{M}_{23}),$$

which means that $\text{Inst}(\mathcal{M}) = \{\langle I_1, I_3\rangle \mid \text{there exists } I_2 \text{ such that } \langle I_1, I_2\rangle \in \text{Inst}(\mathcal{M}_{12}) \text{ and } \langle I_2, I_3\rangle \in \text{Inst}(\mathcal{M}_{23})\}$.

Example 3.2. Let $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ be the schema mappings defined in Example 2.3. Define I_1 by letting $\texttt{Takes}^{I_1} = \{(\text{Alice, Math}), (\text{Alice, Art})\}$. Define I_2 by letting $\texttt{Takes}_1^{I_2} = \texttt{Takes}^{I_1}$ and $\texttt{Student}^{I_2} = \{(\text{Alice, 1234})\}$. Here 1234 is Alice's student id. Define I_3 by letting $\texttt{Enrollment}^{I_3} = \{(1234, \text{Math}), (1234, \text{Art})\}$. It is easy to verify that $\langle I_1, I_2\rangle \in \text{Inst}(\mathcal{M}_{12})$ and that $\langle I_2, I_3\rangle \in \text{Inst}(\mathcal{M}_{23})$. Hence, $\langle I_1, I_3\rangle \in \text{Inst}(\mathcal{M}_{12}) \circ \text{Inst}(\mathcal{M}_{23})$. One of the main problems that we study in this paper is how to find, and in what language, a schema mapping $\mathcal{M} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ that is a composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, according to Definition 3.1. In other words, we will be looking for Σ_{13} (involving only $\mathbf{S}_1$ and $\mathbf{S}_3$) such that an instance $\langle I_1, I_3\rangle$ is in $\text{Inst}(\mathcal{M}_{12}) \circ \text{Inst}(\mathcal{M}_{23})$ if and only if $\langle I_1, I_3\rangle$ satisfies Σ_{13}. A first guess for Σ_{13} in the example we are considering might be the source-to-target tgd

$$\forall n \forall c(\texttt{Takes}(n, c) \rightarrow \exists s \texttt{Enrollment}(s, c)). \tag{1}$$

However, formula (1) does not correctly capture the composition, since in (1), the student id s depends on both the student name n and the course c. But the student id s is supposed to depend only on the student name n (more precisely, (s, c) must be a tuple in the $\texttt{Enrollment}$ relation for every course c where (n, c) is in the $\texttt{Takes}$ relation). In fact, we shall show (in the proof of Proposition 4.4) that in this example, the composition is not definable by any finite set of source-to-target tgds.

Since Inst($\mathcal{M}_{12}$) and Inst($\mathcal{M}_{23}$) are closed under isomorphism, their composition Inst($\mathcal{M}_{12}$) $\circ$ Inst($\mathcal{M}_{23}$) is also closed under isomorphism. Consequently, the class Inst($\mathcal{M}_{12}$)$\circ$Inst($\mathcal{M}_{23}$) can be identified with the following query, which we call the *composition query of* $\mathcal{M}_{12}$ *and* $\mathcal{M}_{23}$: the set of all instances $\langle I_1, I_3\rangle$ such that $\langle I_1, I_3\rangle \in$ Inst($\mathcal{M}_{12}$)$\circ$Inst($\mathcal{M}_{23}$). Note that, according to Definition 3.1, asserting that $\mathcal{M} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ is a composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ amounts to saying that the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is exactly the set of instances over $\langle \mathbf{S}_1, \mathbf{S}_3\rangle$ that satisfy Σ_{13}. In other words, this means that the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is defined by the formulas in the set Σ_{13}.

It is well known and easy to see that every query is definable by an infinitary disjunction of first-order formulas. Specifically, for each finite structure satisfying the query, we construct a first-order formula that defines the structure up to isomorphism and then take the disjunction of all these formulas. This infinitary formula defines the query. Moreover, every query is definable by a set of first-order formulas. Indeed, for each finite structure that does not satisfy the query, we construct the negation of the first-order formula that defines the structure up to isomorphism and then form the set of all such formulas. Note that this is an infinite set of first-order formulas, unless the query is satisfied by all but finitely many non-isomorphic instances. This set is equivalent to its conjunction. Thus, every query is definable by an infinitary conjunction of first-order formulas. It follows that a composition of two schema mappings always exists, since, given two schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, we can obtain a composition $\mathcal{M} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ by taking Σ_{13} to be the singleton consisting of an infinitary formula that defines the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$. Alternatively, we could take Σ_{13} to be the (usually infinite) set of first-order formulas that defines the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$. Since Σ_{13} defines the composition query, this composition Σ_{13} is unique up to logical equivalence in the sense that if $\mathcal{M} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ and $\mathcal{M}' = (\mathbf{S}_1, \mathbf{S}_3, \Sigma'_{13})$ are both compositions of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, then Σ_{13} and Σ'_{13} are logically equivalent. For this reason, from now on we will refer to *the* composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, and will denote it by $\mathcal{M}_{12} \circ \mathcal{M}_{23}$. We may also refer to Σ_{13} as the composition of Σ_{12} and Σ_{23}.

Since the composition query is always definable both by an infinitary formula and by an infinite set of first-order formulas, it is natural to investigate when the composition of two schema mappings is definable in less expressive, but more tractable, logical formalisms. It is also natural to investigate whether the composition of two schema mappings is definable in the same logical formalism that is used to define these two schema mappings. We embark on this investigation in the next section.

4. COMPOSING SOURCE-TO-TARGET TGDS

In this section, we investigate the definability and computational complexity of the composition of two schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ in which the dependencies Σ_{12} and Σ_{23} are finite sets of source-to-target tgds. We shall show the following results.

- If Σ_{12} and Σ_{23} are finite sets of full source-to-target tgds, then the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is also definable by a finite set of full source-to-target tgds.
- If Σ_{12} is a finite set of full source-to-target tgds and Σ_{23} is a finite set of source-to-target tgds (not necessarily full), then the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is definable by a finite set of source-to-target tgds. In turn, this implies that the associated composition query is polynomial-time computable.
- In contrast, if both Σ_{12} and Σ_{23} are finite sets of arbitrary source-to-target tgds (not necessarily full), then the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ may not even be first-order definable, and the associated composition query may be NP-complete.

4.1 Positive Results

Our first positive result shows the good behavior of the composition of mappings, each of which is defined by finite sets of full source-to-target tgds. In the following, whenever α is a formula in which variables $z_1, \ldots, z_l$ may occur, we may use the notation $\alpha[z_1 \mapsto a_1, \ldots, z_l \mapsto a_l]$ to denote the formula obtained by replacing the variables $z_1, \ldots, z_l$ in α by $a_1, \ldots, a_l$, respectively.

Proposition 4.1. *Let $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ be two schema mappings such that Σ_{12} and Σ_{23} are finite sets of full source-to-target tgds. Then, the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is definable by a finite set of full source-to-target tgds. Consequently, the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is a polynomial-time query.*

Proof. Without loss of generality, assume that each full source-to-target tgd in Σ_{12} has a single atom in its right-hand side. We shall show that the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is the schema mapping $(\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$, where Σ_{13} is constructed as follows. For every full source-to-target tgd τ in Σ_{23} of the form $\forall \mathbf{x}((R_1(\mathbf{x_1}) \wedge \cdots \wedge R_k(\mathbf{x_k})) \rightarrow S(\mathbf{x_0}))$, if for some i there is no full source-to-target tgd in Σ_{12} of the form $\forall \mathbf{z_i}(\phi_i \rightarrow R_i(\mathbf{u_i}))$, then no tgd will be constructed from τ. Otherwise, for each i with $1 \leq i \leq k$ and for for each selection of a full source-to-target tgd in Σ_{12} of the form $\forall \mathbf{z_i}(\phi_i \rightarrow R_i(\mathbf{u_i}))$, create a tgd by replacing each atom $R_i(\mathbf{x_i})$ in τ by the formula $\phi_i[\mathbf{u_i} \mapsto \mathbf{x_i}]$. We thereby obtain a full source-to-target tgd from $\mathbf{S}_1$ to $\mathbf{S}_3$ of the form

$$(*) \qquad \forall \mathbf{z}' \forall \mathbf{x}((\phi_1[\mathbf{u_1} \mapsto \mathbf{x_1}] \wedge \cdots \wedge \phi_k[\mathbf{u_k} \mapsto \mathbf{x_k}]) \rightarrow S(\mathbf{x_0})).$$

In the above, $\mathbf{z}'$ includes all the variables in $\phi_1, \ldots, \phi_k$ that are not affected by the replacements. We obtain a finite set Σ_τ of full source-to-target tgds from $\mathbf{S}_1$ to $\mathbf{S}_3$ by allowing each $R_i(\mathbf{x_i})$, for $1 \leq i \leq k$, in τ to be replaced in all possible ways. Then Σ_{13} is the union of all these sets Σ_τ, and it is a finite set, since Σ_{12} and Σ_{23} are both finite sets.

We now show that Σ_{13} gives the composition. We begin by showing that if $\langle I_1, I_3 \rangle$ is in Inst($\mathcal{M}_{12}$) $\circ$ Inst($\mathcal{M}_{23}$), then $\langle I_1, I_3 \rangle$ satisfies Σ_{13}. For every full tgd in Σ_{13} of the form (*), if there exist tuples **a** and **b** of values that replace, correspondingly, the variables in $\mathbf{z}'$ and $\mathbf{x}$, such that

$$(**) \qquad I_1 \models (\phi_1[\mathbf{u_1} \mapsto \mathbf{x_1}] \wedge \cdots \wedge \phi_k[\mathbf{u_k} \mapsto \mathbf{x_k}])\,[\mathbf{z}' \mapsto \mathbf{a}, \mathbf{x} \mapsto \mathbf{b}],$$

we show that $I_3 \models S(\mathbf{x_0})[\mathbf{x} \mapsto \mathbf{b}]$.

By the construction of tgds in Σ_{13}, we know that there are full tgds $\forall \mathbf{z_i}(\phi_i \rightarrow R_i(\mathbf{u_i}))$, for $1 \leq i \leq k$, in Σ_{12}, and a full tgd $\forall \mathbf{x}((R_1(\mathbf{x_1}) \wedge \cdots \wedge R_k(\mathbf{x_k})) \rightarrow S(\mathbf{x_0}))$ in Σ_{23}. We know that there is I_2 such that $\langle I_1, I_2\rangle \models \Sigma_{12}$ and $\langle I_2, I_3\rangle \models \Sigma_{23}$. Since $\langle I_1, I_2\rangle \models \Sigma_{12}$, we obtain from (**) that $I_2 \models R_i(\mathbf{x_i})[\mathbf{x} \mapsto \mathbf{b}]$, for each i with $1 \leq i \leq k$. Since $\langle I_2, I_3\rangle \models \Sigma_{23}$, it then follows that $I_3 \models S(\mathbf{x_0})[\mathbf{x} \mapsto \mathbf{b}]$.

For the converse, assume that $\langle I_1, I_3\rangle$ satisfies Σ_{13}. Let $\langle I_1, I_2\rangle$ be the result of chasing $\langle I_1, \emptyset\rangle$ with the full tgds in Σ_{12}. It is immediate that $\langle I_1, I_2\rangle \models \Sigma_{12}$, by the properties of the chase. We need to show that $\langle I_2, I_3\rangle \models \Sigma_{23}$. Let $\forall \mathbf{x}((R_1(\mathbf{x_1}) \wedge \cdots \wedge R_k(\mathbf{x_k})) \rightarrow S(\mathbf{x_0}))$ be a full tgd in Σ_{23}, and assume that there is a tuple $\mathbf{b}$ of values such that $I_2 \models (R_1(\mathbf{x_1}) \wedge \cdots \wedge R_k(\mathbf{x_k})[\mathbf{x} \mapsto \mathbf{b}]$. We need to show that $I_3 \models S(\mathbf{x_0})[\mathbf{x} \mapsto \mathbf{b}]$.

Since $\langle I_1, I_2\rangle$ is the result of chasing $\langle I_1, \emptyset\rangle$ with the full tgds in Σ_{12}, it follows that there are tgds $\forall \mathbf{z_i}(\phi_i \rightarrow R_i(\mathbf{u_i}))$, with $1 \leq i \leq k$, in Σ_{12}, and a tuple $\mathbf{a}$ of values such that the above condition (**) is true. By the construction of Σ_{13}, we know that a tgd of the form (*) must exist in Σ_{13}. Since $\langle I_1, I_3\rangle$ satisfies this tgd, it follows from the condition (**) that $I_3 \models S(\mathbf{x_0})[\mathbf{x} \mapsto \mathbf{b}]$. This was to be shown. □

A special case of this proposition appeared in Beeri and Vardi [1984b, Lemma 2.3]. An inspection of the proof of Proposition 4.1 shows that the same construction yields the following result.

PROPOSITION 4.2. *Let $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ be two schema mappings such that Σ_{12} is a finite set of full source-to-target tgds and Σ_{23} is a finite set of source-to-target tgds. Then the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is definable by a finite set of source-to-target tgds. Consequently, the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is a polynomial-time query.*

PROOF. The construction of Σ_{13} is the same as in the proof of Proposition 4.1, with the only difference being that for every source-to-target tgd in Σ_{23} of the form $\forall \mathbf{x}((R_1(\mathbf{x_1}) \wedge \cdots \wedge R_k(\mathbf{x_k})) \rightarrow \exists \mathbf{y} S(\mathbf{x_0}, \mathbf{y}))$, and for every i with $1 \leq i \leq k$ and for every full source-to-target tgd in Σ_{12} of the form $\forall \mathbf{z_i}(\phi_i \rightarrow R_i(\mathbf{u_i}))$, we construct a tgd in Σ_{13} of the form:

$$(*) \qquad \forall \mathbf{z}' \forall \mathbf{x}((\phi_1[\mathbf{u_1} \mapsto \mathbf{x_1}] \wedge \cdots \wedge \phi_k[\mathbf{u_k} \mapsto \mathbf{x_k}]) \rightarrow \exists \mathbf{y} S(\mathbf{x_0}, \mathbf{y})).$$

The rest of the proof remains the same as in the proof of Proposition 4.1. □

Example 4.3. We now give an example that shows the use of algorithm of Proposition 4.2. Consider the following three schemas $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$. Schema $\mathbf{S}_1$ consists of a unary relation `EmpAcme` that represents the employees of Acme, a unary relation `EmpAjax` that represents the employees of Ajax, and a unary relation `Local` that represents employees that work in the local office of their company. Schema $\mathbf{S}_2$ consists of a unary relation `Emp` that represents all employees, a unary relation `Local`$_1$ that is intended to be a copy of `Local`, and a unary relation `Over65` that is intended to represent people over age 65. Schema

$\mathbf{S}_3$ consists of a binary relation `Office` that associates employees with office numbers, and a unary relation `CanRetire` that represents employees eligible for retirement. Consider now the schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where

$$\begin{aligned}
\Sigma_{12} \;=\; &\{\forall e(\texttt{EmpAcme}(e) \rightarrow \texttt{Emp}(e)),\\
&\forall e(\texttt{EmpAjax}(e) \rightarrow \texttt{Emp}(e)),\\
&\forall p(\texttt{Local}(p) \rightarrow \texttt{Local}_1(p))\}.\\
\Sigma_{23} \;=\; &\{\forall e(\texttt{Emp}(e) \wedge \texttt{Local}_1(e) \rightarrow \exists o\texttt{Office}(e, o)),\\
&\forall e(\texttt{Emp}(e) \wedge \texttt{Over65}(e) \rightarrow \texttt{CanRetire}(e)\}.
\end{aligned}$$

The result Σ_{13} of applying the composition algorithm from the proof of Proposition 4.2 is

$$\begin{aligned}
\Sigma_{13} \;=\; &\{\forall e(\texttt{EmpAcme}(e) \wedge \texttt{Local}(e) \rightarrow \exists o\texttt{Office}(e, o)),\\
&\forall e(\texttt{EmpAjax}(e) \wedge \texttt{Local}(e) \rightarrow \exists o\texttt{Office}(e, o))\}.
\end{aligned}$$

Note that the first tgd of Σ_{23} is "used twice" (once when we replace `Emp` by `EmpAcme` and once when we replace `Emp` by `EmpAjax`), and the second tgd of Σ_{23} is not used (since there is nothing from $\mathbf{S}_1$ to replace `Over65` by).

It is easy to see that the same result holds for Proposition 4.1 (and Proposition 4.2) when a sequence of more than two consecutive schema mappings is considered. In other words, given a sequence $\mathcal{M}_{12}, \mathcal{M}_{23}, \ldots, \mathcal{M}_{k-1,k}$ of schema mappings where each schema mapping is specified by a finite set of full source-to-target tgds, the composition $\mathcal{M}_{12} \circ \cdots \circ \mathcal{M}_{k-1,k}$ is also definable by a finite set of full source-to-target tgds. If the last schema mapping $\mathcal{M}_{k-1,k}$ is specified by a finite set of source-to-target tgds and all of the others are specified by a finite set of full source-to-target tgds, then the composition $\mathcal{M}_{12} \circ \cdots \circ \mathcal{M}_{k-1,k}$ is definable by a finite set of source-to-target tgds.

4.2 Negative Results

We now present a series of negative results associated with the composition of schema mappings specified by source-to-target tgds.

PROPOSITION 4.4. *There exist schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ such that Σ_{12} is a finite set of source-to-target tgds, Σ_{23} is a finite set of full source-to-target tgds, and the following hold for the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$:*

(1) *$\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is not definable by any finite set of source-to-target tgds, but it is definable by an infinite set of source-to-target tgds.*

(2) *$\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is definable by a first-order formula. Consequently, the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is a polynomial-time query.*

PROOF. The two schema mappings that we use to prove the proposition are the schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ of Example 2.3. Assume that according to the instance I_1, a student with name n is taking courses $c_1, \ldots, c_k$. According to the second tgd of Σ_{12}, this student n is assigned (at least one) student id s.

According to Σ_{23}, the instance I_3 then contains tuples $(s, c_1), \ldots, (s, c_k)$. These requirements are described by the following source-to-target tgd, which we denote by ϕ_k:

$$\forall n \forall c_1 \cdots \forall c_k (\texttt{Takes}(n, c_1) \wedge \cdots \wedge \texttt{Takes}(n, c_k) \rightarrow \\ \exists s (\texttt{Enrollment}(s, c_1) \wedge \cdots \wedge \texttt{Enrollment}(s, c_k)))$$

We next show that the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is given by $\mathcal{M} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$, where Σ_{13} is the infinite set $\{\phi_1, \ldots, \phi_k, \ldots\}$ of source-to-target tgds.

Assume first that $\langle I_1, I_3 \rangle \in \text{Inst}(\mathcal{M}_{12}) \circ \text{Inst}(\mathcal{M}_{23})$. This means that there is I_2 over $\mathbf{S}_2$ such that $\langle I_1, I_2 \rangle \models \Sigma_{12}$ and $\langle I_2, I_3 \rangle \models \Sigma_{23}$. We need to show that $\langle I_1, I_3 \rangle \models \phi_k$, for each $k \geq 1$. Assume that $\texttt{Takes}^{I_1}$ contains tuples $(n, c_1), \ldots, (n, c_k)$, where n is a concrete student name, and $c_1, \ldots, c_k$ are concrete courses. Since $\langle I_1, I_2 \rangle \models \Sigma_{12}$, we obtain that $\texttt{Takes}_1^{I_2}$ contains the tuples $(n, c_1), \ldots, (n, c_k)$ and $\texttt{Student}^{I_2}$ contains the tuple (n, s), for some value s. Since $\langle I_2, I_3 \rangle \models \Sigma_{23}$, we then obtain that $\texttt{Enrollment}^{I_3}$ contains the tuples $(s, c_1), \ldots, (s, c_k)$. Hence, $\langle I_1, I_3 \rangle \models \phi_k$.

Conversely, assume that $\langle I_1, I_3 \rangle \models \phi_k$, for each $k \geq 1$. We need to show that there is I_2 such that $\langle I_1, I_2 \rangle \models \Sigma_{12}$ and $\langle I_2, I_3 \rangle \models \Sigma_{23}$. We construct I_2 as follows'. We let $\texttt{Student}^{I_2}$ be the set of all tuples (n, s) such that: (1) some tuple (n, c) occurs in $\texttt{Takes}^{I_1}$, (2) the set $\{c_1, \ldots, c_l\}$ is the set of *all* courses c such that (n, c) appears in $\texttt{Takes}^{I_1}$, and (3) s is such that $\texttt{Enrollment}^{I_3}$ contains the tuples $(s, c_1), \ldots, (s, c_l)$. We note that s as in condition (3) must exist, whenever $\texttt{Takes}^{I_1}$ contains tuples $(n, c_1), \ldots, (n, c_l)$. This is due to the fact that $\langle I_1, I_3 \rangle$ satisfies ϕ_l. Furthermore, we let $\texttt{Takes}_1^{I_2} = \texttt{Takes}^{I_1}$. It is immediate that $\langle I_1, I_2 \rangle \models \Sigma_{12}$.

We now show that $\langle I_2, I_3 \rangle \models \Sigma_{23}$. Indeed, assume that $\texttt{Student}^{I_2}$ contains a tuple (n, s), and that $\texttt{Takes}_1^{I_2}$ contains a tuple (n, c); we must show that the tuple (s, c) is in $\texttt{Enrollment}^{I_3}$. By construction of $\texttt{Takes}_1^{I_2}$, we know that (n, c) is in $\texttt{Takes}^{I_1}$. Let $\{c_1, \ldots, c_l\}$ be the set of all courses c' such that (n, c') is in $\texttt{Takes}^{I_1}$; this set certainly contains c. By construction of $\texttt{Student}^{I_2}$, we know that s has the property that $\texttt{Enrollment}^{I_3}$ contains the tuples $(s, c_1), \ldots, (s, c_l)$. Since c is a member of $\{c_1, \ldots, c_l\}$, it follows that the tuple (s, c) is in $\texttt{Enrollment}^{I_3}$, as desired.

It can be verified that Σ_{13} is not equivalent to any finite subset of it. We now show that, in fact, Σ_{13} is not equivalent to *any* finite set of source-to-target tgds. The proof of this uses the chase as well as the concept of universal solution. Suppose there is a finite set Σ_{13}^{fin} of source-to-target tgds that is logically equivalent to Σ_{13}. Let $\mathcal{M}^{\text{fin}} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13}^{\text{fin}})$ and consider the following source instance I_1:

$$\texttt{Takes}^{I_1} = \{(n, c_1), \ldots, (n, c_m)\},$$

where n is some student name and $c_1, \ldots, c_m$ are the courses that this student takes. We assume that m is a large enough number, that we shall specify shortly.

We construct an instance I_3 over $\mathbf{S}_3$, by *chasing* (as in Fagin et al. [2005a]) the instance $\langle I_1, \emptyset \rangle$ with the source-to-target tgds in Σ_{13}^{fin}, where $\emptyset$ is an empty instance. The chase applies the source-to-target tgds in Σ_{13}^{fin} and adds into I_3 all the necessary tuples whenever it finds a source-to-target tgd that is not satisfied. This is repeated until all the source-to-target tgds are satisfied. Note that the chase terminates since we are chasing with source-to-target tgds. New values, also called *nulls*, different from the source values in I_1 and different from any other values that may have been added earlier, may appear as part of the tuples added in a chase step with a source-to-target tgd. These nulls are used to replace the existentially quantified variables. Since I_3 is the result of the chase, it follows from a theorem in Fagin et al. [2005a] that I_3 is a universal solution for I_1 under $\mathcal{M}^{\text{fin}}$.

In particular, I_3 is a solution for I_1 under $\mathcal{M}^{\text{fin}}$, that is, $\langle I_1, I_3\rangle \models \Sigma_{13}^{\text{fin}}$. Since Σ_{13}^{fin} and Σ_{13} are equivalent, we have that $\langle I_1, I_3\rangle \models \Sigma_{13}$, and in particular, $\langle I_1, I_3\rangle \models \phi_m$. It follows that $\texttt{Enrollment}^{I_3}$ must contain a set of tuples of the form $(s, c_1), \ldots, (s, c_m)$ for some value s. We now show that s cannot appear among the values of I_1. In other words, we show that s must be a null. For this, we use the fact that I_3 is universal.

Consider the following instance V over $\mathbf{S}_3$: $\texttt{Enrollment}^V = \{(S, c_1), \ldots, (S, c_m)\}$ where S is a null representing a student id. It is easy to see that $\langle I_1, V\rangle \models \Sigma_{13}$. Since Σ_{13}^{fin} and Σ_{13} are equivalent, it follows that $\langle I_1, V\rangle \models \Sigma_{13}^{\text{fin}}$ and, hence, V is a solution for I_1 under $\mathcal{M}^{\text{fin}}$. Since I_3 is a universal solution for I_1 under $\mathcal{M}^{\text{fin}}$, there must exist a homomorphism h from I_3 to V such that $h(v) = v$ for every source value v. But every homomorphism from I_3 to V is forced to map s into the null S. Hence, s cannot be a source value (or, otherwise, $h(s)$ would have to be s). Thus, we showed that $\texttt{Enrollment}^{I_3}$ contains a set $\{(s, c_1), \ldots, (s, c_m)\}$ of tuples where s is a null.

Let l be the maximum number of atoms that are under the scope of existential quantifiers in any source-to-target tgd in Σ_{13}^{fin}. Since I_3 is the result of the chase with Σ_{13}^{fin}, it follows that a null in I_3 can occur in at most l tuples. However, if we take m to be larger than l, then the above obtained set $\{(s, c_1), \ldots, (s, c_m)\}$ of tuples gives a contradiction. Therefore, Σ_{13} is not logically equivalent to any finite set of source-to-target tgds.

Finally, the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ of the two schema mappings is definable by the first-order formula $\forall n \exists s \forall c (\texttt{Takes}(n, c) \rightarrow \texttt{Enrollment}(s, c))$. We shall verify this in Example 5.1, where we show that a logically equivalent formula defines the composition. □

It is an interesting open problem to consider the complexity of deciding, given schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, each defined by finite sets of source-to-target tgds, whether the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is definable by a finite set of source-to-target tgds. In particular, it is not even clear whether this problem is decidable.

We have just given an example in which the composition is definable by an infinite set of source-to-target tgds, but it is not definable by any finite set of source-to-target tgds. There is also a different example in which the composition is not definable even by an infinite set of source-to-target tgds. This is stated in the next result, which amplifies the limitations of the language of source-to-target tgds with respect to composition. A proof appears in Section 5.1, after we develop the necessary machinery.

PROPOSITION 4.5. *There exist schema mappings* $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ *and* $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ *such that* Σ_{12} *consists of a single source-to-target tgd,* Σ_{23} *is* *a finite set of full source-to-target tgds, and the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ cannot be defined by any finite or infinite set of source-to-target tgds.*

In the example given in Proposition 4.4, the composition query is polynomial-time computable, since it is first-order. In what follows, we will show that there are schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ such that Σ_{12} is a finite set of source-to-target tgds, Σ_{23} consists of a single full source-to-target tgd, but the composition query for $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is NP-complete. Furthermore, this composition query is not definable by any formula of the finite-variable infinitary logic $L^{\omega}_{\infty\omega}$, which is a powerful formalism that subsumes least fixed-point logic LFP (hence, it subsumes first-order logic and Datalog) on finite structures (see Abiteboul et al. [1995]).

THEOREM 4.6. *There exist schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ such that Σ_{12} is a finite set of source-to-target tgds, each having at most one existential quantifier, Σ_{23} consists of one full source-to-target tgd, and such that the following hold for the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$:*

(1) *The composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is* NP*-complete.*
(2) *The composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is not definable by any formula of $L^{\omega}_{\infty\omega}$, and hence of least fixed-point logic* LFP.

PROOF. Later (Proposition 4.8), we shall show that the composition query of schema mappings definable by finite sets of source-to-target tgds is always in NP. As we now describe, NP-hardness can be obtained by a reduction of 3-COLORABILITY to the composition query of two fixed schema mappings. The schema $\mathbf{S}_1$ consists of a single binary relation symbol E, the schema $\mathbf{S}_2$ consists of two binary relation symbols C and F, and the schema $\mathbf{S}_3$ consists of one binary relation symbol D. The set Σ_{12} consists of the following three source-to-target tgds:

$$\forall x \forall y (E(x, y) \to \exists u C(x, u))$$
$$\forall x \forall y (E(x, y) \to \exists u C(y, u))$$
$$\forall x \forall y (E(x, y) \to F(x, y)).$$

Intuitively, $C(x, u)$ means that node x has color u. The third tgd of Σ_{12} intuitively copies the edge relation E into the relation F. Finally, Σ_{23} consists of a single full source-to-target tgd:

$$\forall x \forall y \forall u \forall v (C(x, u) \wedge C(y, v) \wedge F(x, y) \to D(u, v)).$$

Intuitively, this tgd says that if u and v are the colors of adjacent nodes, then the tuple (u, v) is in the "distinctness" relation D, which we shall take to consist of tuples of distinct colors. Thus, if u and v are the colors of adjacent nodes, then we are forcing u and v to be distinct colors.

Let I_3 be the instance over the schema $\mathbf{S}_3$ with

$$D^{I_3} = \{(r, g), (g, r), (b, r), (r, b), (g, b), (b, g)\}.$$

In words, D^{I_3} contains all pairs of different colors among the three colors r, g, and b. Let $\mathbf{G} = (V, E)$ be a graph and let I_1 be the instance over $\mathbf{S}_1$

consisting of the edge relation E of $\mathbf{G}$. We claim that $\mathbf{G}$ is 3-colorable if and only if $\langle I_1, I_3\rangle \in \text{Inst}(\mathcal{M}_{12}) \circ \text{Inst}(\mathcal{M}_{23})$. This is sufficient to prove the theorem, since 3-COLORABILITY is NP-complete [Garey et al. 1976].

If $\mathbf{G}$ is 3-colorable, then there is a function c from V to the set $\{r, b, g\}$ such that for every edge $(x, y) \in E$, we have that $c(x) \neq c(y)$. Let I_2 be the instance over $\mathbf{S_2}$ with $C^{I_2} = \{(x, c(x)) : x \in V\}$ and $F^{I_2} = E$. Clearly, $\langle I_1, I_2\rangle \in \text{Inst}(\mathcal{M}_{12})$ and $\langle I_2, I_3\rangle \in \text{Inst}(\mathcal{M}_{23})$. Therefore, $\langle I_1, I_3\rangle \in \text{Inst}(\mathcal{M}_{12}) \circ \text{Inst}(\mathcal{M}_{23})$.

Conversely, assume that $\langle I_1, I_3\rangle$ is in $\text{Inst}(\mathcal{M}_{12}) \circ \text{Inst}(\mathcal{M}_{23})$. This means there exists an instance I_2 over $\mathbf{S}_2$ such that $\langle I_1, I_2\rangle \in \text{Inst}(\mathcal{M}_{12})$ and $\langle I_2, I_3\rangle \in \text{Inst}(\mathcal{M}_{23})$. The first two source-to-target tgds in Σ_{12} state that for each node n incident to an edge there exists some u such that $C(n, u)$, while the third source-to-target tgd in Σ_{12} asserts that the edge relation E is contained in F^{I_2}. We construct a coloring function c as follows. For each node n that is incident to an edge we take $c(n) = u$, where u is picked arbitrarily among those u that satisfy $C(n, u)$. Since D^{I_3} is the inequality relation on $\{r, g, b\}$, the full source-to-target tgd in Σ_{23} enforces that for every edge of $\mathbf{G}$, and no matter which u we picked for a given n, the two vertices of that edge are assigned different colors among the three colors r, g and b. Therefore, $\mathbf{G}$ is 3-colorable, as desired.

The above reduction of 3-COLORABILITY to the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ belongs to a class of weak polynomial-time reductions known as *quantifier-free* reductions, since the instance $\langle I_1, I_3\rangle$ of the composition query can be defined from the instance $\mathbf{G} = (V, E)$ using quantifier-free formulas (see Immerman [1999] for the precise definitions). Dawar [1998] showed that 3-COLORABILITY is not expressible in the finite-variable infinitary logic $L^{\omega}_{\infty\omega}$. Since definability in $L^{\omega}_{\infty\omega}$ is preserved under quantifier-free reductions, it follows that the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is not expressible in $L^{\omega}_{\infty\omega}$. In turn, this implies that the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is not expressible in least fixed-point logic LFP, since $L^{\omega}_{\infty\omega}$ subsumes LFP on the class of all finite structures (see Ebbinghaus and Flum [1999]). □

Proposition 4.2 and Theorem 4.6 yield a sharp boundary on the definability of the composition of schema mappings specified by finite sets of source-to-target tgds. Specifically, the composition of a finite set of full source-to-target tgds with a finite set of source-to-target tgds is always definable by a first-order formula (and, in fact, definable by a finite conjunction of source-to-target tgds), while the composition of a finite set of source-to-target tgds, each having at most one existential quantifier, with a set consisting of one full source-to-target tgd may not even be $L^{\omega}_{\infty\omega}$-definable. Similarly, the computational complexity of the associated composition query may jump from solvable in polynomial time to NP-complete.

The HOMOMORPHISM PROBLEM over the schema $\mathbf{S}$ is the following decision problem: given two instances I and J of $\mathbf{S}$, is there a homomorphism from I to J? (Recall that a homomorphism from I to J is a function h such that for every relation symbol R in $\mathbf{S}$ and every tuple $(a_1, \ldots, a_n) \in R^I$, we have that $(h(a_1), \ldots, h(a_n)) \in R^J$.) This is a fundamental algorithmic problem because, as shown by Feder and Vardi [1998], all constraint satisfaction problems can be identified with homomorphism problems. In particular, 3-SAT and 3-COLORABILITY are special cases of the HOMOMORPHISM PROBLEM over suitable schemas. For instance, 3-COLORABILITY amounts to the following problem: given a graph $\mathbf{G}$, is there a homomorphism from $\mathbf{G}$ to the complete 3-node graph $\mathbf{K}_3$? A slight modification of the proof of the preceding Theorem 4.6 shows that for every schema $\mathbf{S}$, the HOMOMORPHISM PROBLEM over $\mathbf{S}$ has a simple quantifier-free reduction to the composition query of two schema mappings specified by finite sets of source-to-target tgds.

PROPOSITION 4.7. *For every schema $\mathbf{S} = \langle R_1, \ldots, R_m \rangle$, there are schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ such that Σ_{12} is a finite set of source-to-target tgds and Σ_{23} is a finite set of full source-to-target tgds, with the property that the* HOMOMORPHISM PROBLEM *over $\mathbf{S}$ has a quantifier-free reduction to the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$.*

PROOF. The schema $\mathbf{S}_1$ is the same as the schema $\mathbf{S} = \langle R_1, \ldots, R_m \rangle$. The schema $\mathbf{S_2}$ is $\langle H, T_1, \ldots, T_m \rangle$, where H is a binary relation symbol and each T_i has the same arity as R_i, for $1 \le i \le m$. The schema $\mathbf{S_3}$ is $\langle P_1, \ldots, P_m \rangle$, where each P_i has the same arity as R_i, for $1 \le i \le m$. The dependencies in Σ_{12} and Σ_{23} are as follows:

$$\begin{aligned}
\Sigma_{12} = \{ &\forall x_1 \cdots \forall x_{k_1} (R_1(x_1, \ldots, x_{k_1}) \to \exists y_1 \cdots \exists y_{k_1} (H(x_1, y_1) \wedge \cdots \wedge H(x_{k_1}, y_{k_1}))),\\
&\vdots\\
&\forall x_1 \cdots \forall x_{k_m} (R_m(x_1, \ldots, x_{k_m}) \to \exists y_1 \cdots \exists y_{k_m} (H(x_1, y_1) \wedge \cdots \wedge H(x_{k_m}, y_{k_m}))),\\
&\forall \mathbf{x}(R_1(\mathbf{x}) \to T_1(\mathbf{x})),\\
&\vdots\\
&\forall \mathbf{x}(R_m(\mathbf{x}) \to T_m(\mathbf{x}))\}
\end{aligned}$$

$$\begin{aligned}
\Sigma_{23} = \{ &\forall x_1 \forall y_1 \cdots \forall x_{k_1} \forall y_{k_1}\\
&\quad ((H(x_1, y_1) \wedge \cdots \wedge H(x_{k_1}, y_{k_1}) \wedge T_1(x_1, \ldots, x_{k_1})) \to P_1(y_1, \ldots, y_{k_1})),\\
&\vdots\\
&\forall x_1 \forall y_1 \cdots \forall x_{k_m} \forall y_{k_m}\\
&\quad ((H(x_1, y_1) \wedge \cdots \wedge H(x_{k_m}, y_{k_m}) \wedge T_m(x_1, \ldots, x_{k_m})) \to P_m(y_1, \ldots, y_{k_m}))\}.
\end{aligned}$$

Intuitively, the R_i relation is being copied into the T_i relation, for $1 \le i \le m$, and $H(x, y)$ means that a homomorphism is mapping x to y.

Let $I = \langle R_1^I, \ldots, R_m^I \rangle$ and $J = \langle R_1^J, \ldots, R_m^J \rangle$ be two instances of $\mathbf{S}$. Since $\mathbf{S}_1$ is the same as $\mathbf{S}$, we have that I is an instance of $\mathbf{S}_1$. Let J' be the instance over $\mathbf{S}_3$ where $P_i^{J'} = R_i^J$, for $1 \le i \le m$. (Thus, J' is the same as J except that the relation names reflect schema $\mathbf{S}_3$ rather than $\mathbf{S}_1$.) It is easy to verify that there is a homomorphism from I to J if and only if $\langle I, J' \rangle$ is in $\text{Inst}(\mathcal{M}_{12}) \circ \text{Inst}(\mathcal{M}_{23})$. □

The next result establishes an upper bound on the computational complexity of the composition query associated with two schema mappings specified by finite sets of source-to-target tgds. It also shows that the composition of two such mappings is always definable by an existential second-order formula.

PROPOSITION 4.8. *If $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ are schema mappings such that Σ_{12} and Σ_{23} are finite sets of source-to-target tgds, then the* *composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is in* NP. *Consequently, the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is definable by an existential second-order formula.*

Proof. To establish membership in NP, it suffices to show that if $\langle I_1, I_3 \rangle \in \text{Inst}(\mathcal{M}_{12}) \circ \text{Inst}(\mathcal{M}_{23})$, then there is an instance I_2 over $\mathbf{S}_2$ that has size polynomial in the sizes of I_1 and I_3, and is such that $\langle I_1, I_2 \rangle \models \Sigma_{12}$ and $\langle I_2, I_3 \rangle \models \Sigma_{23}$.

Suppose we have I_1 and I_3 as above. Then there is an instance J such that $\langle I_1, J \rangle \models \Sigma_{12}$ and $\langle J, I_3 \rangle \models \Sigma_{23}$. Since Σ_{12} is a set of source-to-target tgds, the schema mapping $\mathcal{M}_{12}$ is a data exchange setting with source $\mathbf{S}_1$ and target $\mathbf{S}_2$ (and no target dependencies). Moreover, by results of Fagin et al. [2005a], in this data exchange setting there is a *universal* solution U for I_1 of size polynomial in the size of I_1. By definition, a universal solution U for I_1 has the property that, for every solution for I_1, there is a homomorphism h from U to that solution such that h is the identity on values from I_1. In particular, there is a homomorphism $h : U \to J$ such that $h(v) = v$, for every value v from I_1. Let $I_2 = h(U)$. Clearly, I_2 is an instance over $\mathbf{S_2}$, has size at most the size of U, and is a subinstance of J. Since (a) Σ_{12} is a set of source-to-target tgds, (b) $\langle I_1, U \rangle \models \Sigma_{12}$, and (c) h is a homomorphism from U to I_2 that is the identity on values from I_1, we have that $\langle I_1, I_2 \rangle \models \Sigma_{12}$. Furthermore, since (a) Σ_{23} is a set of source-to-target tgds, (b) $\langle J, I_3 \rangle \models \Sigma_{23}$, and (c) I_2 is a subinstance of J, we have that $\langle I_2, I_3 \rangle \models \Sigma_{23}$.

The fact that the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is in NP implies, by Fagin's Theorem [Fagin 1974], that the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is definable on instances $\langle I_1, I_3 \rangle$ over $\langle \mathbf{S}_1, \mathbf{S}_3 \rangle$ by an existential second-order formula, where the existential second-order variables are interpreted over relations on the union of the set of values in I_1 with the set of values in I_3. □

We conclude this section by showing that the results of Proposition 4.8 may fail dramatically for schema mappings specified by arbitrary first-order formulas.

Proposition 4.9. *There are schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ such that Σ_{12} consists of a single first-order formula, Σ_{23} is the empty set, and the composition query of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is undecidable.*

Proof. We define $\mathcal{M}_{12}$ in such a way that $\langle I_1, I_2 \rangle \in \text{Inst}(\mathcal{M}_{12})$ precisely when I_1 is the encoding of a Turing machine and I_2 represents a terminating computation of that Turing machine (thus, Σ_{12} consists of a first-order formula that expresses this connection). We let the schema $\mathbf{S}_3$ consist of, say, a single unary relation symbol, and let Σ_{23} be the empty set. So, the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ consists of all $\langle I_1, I_3 \rangle$ where I_1 is the encoding of a halting Turing machine, and I_3 is arbitrary. The result follows from the fact that it is undecidable to determine if a Turing machine is halting. □

5. SECOND-ORDER TGDS

We have seen in the previous section that the composition of two schema mappings specified by finite sets of source-to-target tgds may not be definable by a set (finite or infinite) of source-to-target tgds. From Proposition 4.8, however, we know that such a composition is always definable by an existential second-order formula. We shall show in this section that, in fact, the composition of schema mappings, each specified by a finite set of source-to-target tgds, is always definable by a restricted form of existential second-order formula, which we call a *second-order tuple-generating dependency (SO tgd)*. Intuitively, an SO tgd is a source-to-target tgd suitably extended with existentially quantified functions and with equalities. Every finite set of source-to-target tgds is equivalent to an SO tgd. Furthermore, an SO tgd is capable of defining the composition of two schema mappings that are specified by SO tgds. In other words, SO tgds are *closed under composition*. Moreover, we shall show in Section 6 that SO tgds possess good properties for data exchange. All these properties justify SO tgds as the right language for representing schema mappings and for composing schema mappings.

Example 5.1. The proof of Proposition 4.4 shows that for the two schema mappings of Example 2.3 there is no finite set of source-to-target tgds that can define the composition. At the end of the proof of Proposition 4.4, it was noted that the composition is defined by the first-order formula $\forall n \exists s \forall c(\texttt{Takes}(n, c) \rightarrow \texttt{Enrollment}(s, c))$. If we Skolemize this formula, we obtain the following formula, which is an SO tgd that defines the composition:

$$\exists f\, (\forall n \forall c\, (\texttt{Takes}(n, c) \rightarrow \texttt{Enrollment}(f(n), c))) \tag{2}$$

In this formula, f is a function symbol that associates each student name n with a student id $f(n)$. The SO tgd states that whenever a student name n is associated with a course c in Takes, then the corresponding student id $f(n)$ is associated with c in Enrollment. This is independent of how many courses a student takes: if student name n is associated with courses $c_1, \ldots, c_k$ in Takes, then $f(n)$ is associated with all of $c_1, \ldots, c_k$ in Enrollment.

We now verify that (2) does indeed define the composition. Assume first that $\langle I_1, I_3\rangle \in \text{Inst}(\mathcal{M}_{12}) \circ \text{Inst}(\mathcal{M}_{23})$. Then there is I_2 over $\mathbf{S}_2$ such that $\langle I_1, I_2\rangle \models \Sigma_{12}$ and $\langle I_2, I_3\rangle \models \Sigma_{23}$. We construct a function f^0 as follows. For each n such that (n, c) is in $\texttt{Takes}^{I_1}$, we set $f^0(n) = s$, where s is such that (n, s) is in $\texttt{Student}^{I_2}$ (such s is guaranteed to exist according to the second source-to-target tgd in Σ_{12}, and we pick one such s). It is immediate that $\langle I_1, I_3\rangle$ satisfies the SO tgd when the existentially quantified function symbol f is instantiated with the constructed f^0. Conversely, assume that $\langle I_1, I_3\rangle$ satisfies the SO tgd. Then there is a function f^0 such that for every (n, c) in $\texttt{Takes}^{I_1}$ we have that $(f^0(n), c)$ is in $\texttt{Enrollment}^{I_3}$. Let I_2 be such that $\texttt{Student}^{I_2} = \{(n,\ f^0(n) \mid (n, c) \in \texttt{Takes}^{I_1}\}$ and $\texttt{Takes}_1^{I_2} = \texttt{Takes}^{I_1}$. It can be verified that $\langle I_1, I_2\rangle \models \Sigma_{12}$ and $\langle I_2, I_3\rangle \models \Sigma_{23}$.

Example 5.2. This example illustrates a slightly more complex form of a second-order tgd that contains equalities between terms. Consider the following three schemas $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$. Schema $\mathbf{S}_1$ consists of a single unary relation symbol Emp of employees. Schema $\mathbf{S}_2$ consists of a single binary relation symbol $\texttt{Mgr}_1$, that associates each employee with a manager. Schema $\mathbf{S}_3$ consists of a similar binary relation symbol Mgr, that is intended to provide a copy of $\texttt{Mgr}_1$. and an additional unary relation symbol SelfMgr, that is intended to store employees who are their own manager. Consider now the schema mappings

$\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where

$$\Sigma_{12} = \{\forall e\,(\texttt{Emp}(e) \rightarrow \exists m \texttt{Mgr}_1(e, m))\} \quad \Sigma_{23} = \{\forall e \forall m\,(\texttt{Mgr}_1(e, m) \rightarrow \texttt{Mgr}(e, m)), \\ \forall e(\texttt{Mgr}_1(e, e) \rightarrow \texttt{SelfMgr}(e))\}.$$

It is straightforward to verify that the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is $\mathcal{M}_{13}$, where Σ_{13} is the following second-order tgd:

$$\exists f\,(\forall e(\texttt{Emp}(e) \rightarrow \texttt{Mgr}(e, f(e))) \wedge \\ \forall e(\texttt{Emp}(e) \wedge (e = f(e)) \rightarrow \texttt{SelfMgr}(e))).$$

In fact, we shall derive this later when we give a composition algorithm.

We will use this example in Section 5.1 to show that equalities in SO tgds are strictly necessary for the purposes of composition, and also to give a proof for the earlier Proposition 4.5.

Before we formally define SO tgds, we need to define *terms*. Given a collection $\mathbf{x}$ of variables and a collection $\mathbf{f}$ of function symbols, a *term (based on* $\mathbf{x}$ *and* $\mathbf{f}$*)* is defined recursively as follows:

(1) Every variable in $\mathbf{x}$ is a term.
(2) If f is a k-ary function symbol in $\mathbf{f}$ and $t_1, \ldots, t_k$ are terms, then $f(t_1, \ldots, t_k)$ is a term.

We now give the precise definition of an SO tgd.[1]

Definition 5.3. Let $\mathbf{S}$ be a source schema and $\mathbf{T}$ a target schema. A *second-order tuple-generating dependency (SO tgd)* is a formula of the form:

$$\exists \mathbf{f}((\forall \mathbf{x_1}(\phi_1 \rightarrow \psi_1)) \wedge \cdots \wedge (\forall \mathbf{x_n}(\phi_n \rightarrow \psi_n))), \tag{3}$$

where

(1) Each member of $\mathbf{f}$ is a function symbol.
(2) Each ϕ_i is a conjunction of
    - atomic formulas of the form $S(y_1, \ldots, y_k)$, where S is a k-ary relation symbol of schema $\mathbf{S}$ and $y_1, \ldots, y_k$ are variables in $\mathbf{x_i}$, not necessarily distinct, and
    - equalities of the form $t = t'$ where t and t' are terms based on $\mathbf{x_i}$ and $\mathbf{f}$.
(3) Each ψ_i is a conjunction of atomic formulas $T(t_1, \ldots, t_l)$, where T is an l-ary relation symbol of schema $\mathbf{T}$ and $t_1, \ldots, t_l$ are terms based on $\mathbf{x_i}$ and $\mathbf{f}$.
(4) Each variable in $\mathbf{x_i}$ appears in some atomic formula of ϕ_i.

We may refer to each subformula $\forall \mathbf{x_i}(\phi_i \rightarrow \psi_i)$ as a *conjunct* of the second-order tgd; we may also use the shorthand notation C_i for this conjunct.

The fourth condition is a "safety" condition, analogous to that made for (first-order) source-to-target tgds. As an example, the following formula is not a valid

[1] This definition is slightly different from that given in our conference version [Fagin et al. 2004]. Every SO tgd as defined here is an SO tgd as defined in Fagin et al. [2004], but not conversely. However, every SO tgd as defined in Fagin et al. [2004] is logically equivalent to an SO tgd as defined here.

second-order tgd:

$$\exists f \exists g \forall x \forall y (S(x) \wedge (g(y) = f(x)) \rightarrow T(x, y)).$$

The safety condition is violated, since the variable y does not appear in an atomic formula on the left-hand side.

There is a subtlety in the definition of SO tgds, namely, the semantics of existentialized function symbols.[2] What should the domain and range of the corresponding functions be? Thus, if we are trying to evaluate whether the SO tgd (3) is satisfied by $\langle I, J \rangle$, what should the domain and range be for the concrete functions that may replace the existentialized function symbols in $\mathbf{f}$? Perhaps the most obvious choice is to let the domain and range be the active domain of $\langle I, J \rangle$ (the *active domain* of $\langle I, J \rangle$ consists of those values that appear in I and/or J). In the proof of Proposition 4.8, the existential second-order variables are interpreted over relations on the active domain. But as we shall see in Section 7.3, this choice of the active domain as the universe may give us the "wrong answer". Intuitively, if our instance $\langle I, J \rangle$ is $\langle I_1, I_3 \rangle$, we may wish the functions to take on values in the "missing middle instance" I_2, which may be much bigger than I_1 and I_3.

We define the semantics by converting each instance $\langle I, J \rangle$ into a structure $\langle U; I, J \rangle$, which is just like $\langle I, J \rangle$ except that it has a *universe* U. The domain and range of the functions is then taken to be U. We take the universe U to be a countably infinite set that includes the active domain. The intuition is that the universe contains the active domain along with an infinite set of nulls. Then, if σ is an SO tgd, we define $\langle I, J \rangle \models \sigma$ to hold precisely if $\langle U; I, J \rangle \models \sigma$ under the standard notion of satisfaction in second-order logic (see, e.g., Ebbinghaus and Flum [1999] or Enderton [2001]). The standard notion of satisfaction says that if σ is $\exists \mathbf{f} \sigma'$, where σ' is first-order, then $\langle U; I, J \rangle \models \sigma$ precisely if there is a collection $\mathbf{f^0}$ of functions with domain and range U such that $\langle U; I, J \rangle$ satisfies σ' when each function symbol in $\mathbf{f}$ is replaced by the corresponding function in $\mathbf{f^0}$. We may write $\langle U; I, J \rangle \models \sigma'[\mathbf{f} \mapsto \mathbf{f^0}]$ to represent this situation, or simply $\langle I, J \rangle \models \sigma'[\mathbf{f} \mapsto \mathbf{f^0}]$ when the universe U is fixed and understood from the context. As we shall see in Section 5.2, instead of taking the universe U to be infinite, we can take it to be finite and "sufficiently large".

Several remarks are in order now. First, SO tgds are closed under conjunction. That is, if σ_1 and σ_2 are SO tgds, then the conjunction $\sigma_1 \wedge \sigma_2$ is logically equivalent to an SO tgd. This is because we simply rename the function symbols in σ_2 to be disjoint from those in σ_1; then, if σ_1 is $\exists \mathbf{f}_1 \sigma_1'$, and σ_2 is $\exists \mathbf{f}_2 \sigma_2'$, with $\mathbf{f}_1$ and $\mathbf{f}_2$ disjoint, the conjunction $\exists \mathbf{f}_1 \sigma_1' \wedge \exists \mathbf{f}_2 \sigma_2'$ is logically equivalent to $\exists \mathbf{f}_1 \exists \mathbf{f}_2 (\sigma_1' \wedge \sigma_2')$. Of course, the fact that SO tgds are closed under conjunction implies that every finite set of SO tgds is logically equivalent to a single SO tgd. For this reason, when we consider schema mappings specified by SO tgds, it is enough to restrict our attention to the case where the set Σ_{st} consists of one SO tgd. We will then identify the singleton set Σ_{st} with the SO tgd itself, and refer to Σ_{st} as an SO tgd.

[2] This subtlety was pointed out to us by Sergey Melnik, in the context of domain independence (which we shall discuss in Section 5.2).

Second, it should not come as a surprise that every (first-order) source-to-target tgd is equivalent to an SO tgd. In fact, it is easy to see that every source-to-target tgd is equivalent to an SO tgd without equalities. Specifically, let σ be the source-to-target tgd

$$\forall x_1 \cdots \forall x_m (\phi_S(x_1, \ldots, x_m) \to \exists y_1 \cdots \exists y_n \psi_T(x_1, \ldots, x_m, y_1, \ldots, y_n)).$$

Then σ is equivalent to the following SO tgd without equalities, which is obtained by Skolemizing σ:

$$\exists f_1 \cdots \exists f_n \forall x_1 \cdots \forall x_m (\phi_S(x_1, \ldots, x_m) \to \\ \psi_T(x_1, \ldots, x_m, f_1(x_1, \ldots, x_m), \ldots, f_n(x_1, \ldots, x_m))).$$

Given a finite set Σ of source-to-target tgds, we can find an SO tgd that is equivalent to Σ by taking, for each tgd σ in Σ, a conjunct of the SO tgd to capture σ as described above (we use disjoint sets of function symbols in each conjunct, as before).

Third, we point out that every SO tgd is equivalent to an SO tgd in a "normal form" where the right-hand sides (i.e., the formulas ψ_i in (3)) are atomic formulas, rather than conjunctions of atomic formulas. For example, consider the SO tgd

$$\exists f \forall x (R(x) \to (S(x, f(x)) \wedge T(f(x), x))).$$

This SO tgd is logically equivalent to the SO tgd

$$\exists f (\forall x (R(x) \to S(x, f(x))) \wedge \forall x (R(x) \to T(f(x), x))).$$

This is unlike the situation for (first-order) source-to-target dependencies, where we would lose expressive power if we required that the right-hand sides consist only of atomic formulas and not conjunctions of atomic formulas. In our composition algorithm that we shall present in Section 7, we begin by converting SO tgds to this normal form.

The next three subsections delve into further details on second-order tgds. We first show that equalities are strictly needed in the definition of SO tgds (or else we lose expressive power). We then show that the choice of the universe for SO tgds does not really matter, as long as the universe contains the active domain and is sufficiently large. The section concludes with a consideration of the model-checking problem and how it differs from the first-order case.

5.1 The Necessity of Equalities in Second-Order TGDs

Our definition of SO tgds allows for equalities between terms in the formulas ϕ_i, even though we just saw that SO tgds that represent first-order tgds do not require equalities. The next theorem (or its corollary) tells us that such equalities are necessary, since it may not be possible to define the composition of two schema mappings otherwise. This theorem is stated in more generality than simply saying that equalities are necessary, in order to provide a proof of Proposition 4.5.

THEOREM 5.4. *There exist schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ where Σ_{12} consists of a single source-to-target tgd, Σ_{23} is a finite set of full source-to-target tgds, and the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is given by an SO* *tgd that is not logically equivalent to any finite or infinite set of SO tgds without equalities.*

Proof. Let $\mathbf{S}_1$, $\mathbf{S}_2$, $\mathbf{S}_3$, Σ_{12}, Σ_{23}, and Σ_{13} be as in Example 5.2. We need only show that there is no finite or infinite set of SO tgds without equalities that is logically equivalent to Σ_{13}.

Define I_1 by letting $\texttt{Emp}^{I_1} = \{\text{Bob}\}$. Define I_3 by letting $\texttt{Mgr}^{I_3} = \{(\text{Bob, Susan})\}$ and $\texttt{SelfMgr}^{I_3} = \emptyset$. Define I'_3 by letting $\texttt{Mgr}^{I'_3} = \{(\text{Bob, Bob})\}$ and $\texttt{SelfMgr}^{I'_3} = \emptyset$. It is easy to see that $\langle I_1, I_3\rangle \models \Sigma_{13}$; intuitively, we let $f(\text{Bob}) = \text{Susan}$. It is also easy to see that $\langle I_1, I'_3\rangle \not\models \Sigma_{13}$, since $\texttt{SelfMgr}^{I'_3}$ does not contain Bob.

We shall show that every SO tgd without equalities that is satisfied by $\langle I_1, I_3\rangle$ is also satisfied by $\langle I_1, I'_3\rangle$. Since also $\langle I_1, I_3\rangle \models \Sigma_{13}$ but $\langle I_1, I'_3\rangle \not\models \Sigma_{13}$, it follows easily that Σ_{13} is not equivalent to any finite or infinite set of SO tgds without equalities, which proves the theorem.

Let σ be an SO tgd without equalities that is satisfied by $\langle I_1, I_3\rangle$. The proof is complete if we show that σ is satisfied by $\langle I_1, I'_3\rangle$. Assume that σ is

$$\exists \mathbf{f}((\forall \mathbf{x_1}(\phi_1 \to \psi_1)) \wedge \cdots \wedge (\forall \mathbf{x_n}(\phi_n \to \psi_n))).$$

We begin by showing that $\texttt{SelfMgr}$ does not appear in σ. Assume that $\texttt{SelfMgr}$ appears in σ; we shall derive a contradiction. By the definition of an SO tgd, we know that there is i and some term t such that $\texttt{SelfMgr}(t)$ appears in ψ_i. Since by assumption ϕ_i does not contain any equalities, it follows that ϕ_i contains only formulas of the form $\texttt{Emp}(x)$, with x a member of $\mathbf{x_i}$. So ϕ_i can be satisfied in I_1, by letting Bob play the role of all of the variables in $\mathbf{x_i}$. Since $\texttt{SelfMgr}^{I_3}$ is empty, it follows that ψ_i is not satisfied under this (or any) assignment. Therefore, σ is not satisfied in $\langle I_1, I_3\rangle$, which is the desired contradiction.

We conclude the proof by showing that $\langle I_1, I'_3\rangle$ satisfies σ. Let the role of every function symbol in $\mathbf{f}$ be played by a constant function (of the appropriate arity) that always takes on the value Bob. Consider a conjunct $\forall \mathbf{x_i}(\phi_i \to \psi_i)$ of σ. We must show that if ϕ_i holds in I_1 for some assignment to the variables in $\mathbf{x_i}$, then ψ_i holds in I'_3 for the same assignment. It follows from the fourth condition (the safety condition) in the definition of SO tgds that the conjuncts of ϕ_i are precisely all formulas of the form $\texttt{Emp}(x)$ for x in $\mathbf{x_i}$. Since ϕ_i holds in I_1, every variable x in $\mathbf{x_i}$ is assigned the value Bob. Therefore, every term in ψ_i is assigned the value Bob. Since by assumption ψ_i does not contain $\texttt{SelfMgr}$, it follows that every conjunct in ψ_i is of the form $\texttt{Mgr}(t_1, t_2)$. Since, as we just showed, t_1 and t_2 are both assigned the value Bob, it follows that ψ_i holds in I'_3. This was to be shown. □

Our desired result about the necessity of equalities in SO tgds is an immediate corollary.

Corollary 1. *There exist schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ where Σ_{12} and Σ_{23} are finite sets of source-to-target tgds, and the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is given by an SO tgd that is not equivalent to any SO tgd without equalities.*

We consider it quite interesting that allowing equalities in SO tgds is necessary to make them sufficiently expressive. This is particularly true because the "obvious" way to define SO tgds does not allow equalities. Indeed, as we saw, when we Skolemize a source-to-target tgd to obtain an SO tgd, no equalities are introduced.

Because of the generality with which we stated Theorem 5.4, we obtain a proof of Proposition 4.5.

PROOF OF PROPOSITION 4.5. This follows immediately from Theorem 5.4 and the fact, noted earlier, that every source-to-target tgd is equivalent to an SO tgd without equalities. □

5.2 The Choice of Universe

In our definition of the semantics of SO tgds, we took the universe (which serves as the domain and range of the existentially quantified functions) to be a countably infinite set that includes the active domain. In this section, we show that if instead of taking the universe to be infinite, we take it to be finite but sufficiently large, then the semantics is unchanged. We also show that the choice of the universe does not matter, as long as the universe contains the active domain and is large enough.

Before we state and prove this theorem about the choice of the universe, we need another definition, that we will make use of several times in this article. Let $\mathbf{x}$ be a collection of variables and $\mathbf{f}$ a collection of function symbols. Similarly to our earlier definition of terms, a *term* (*based on* $\mathbf{x}$ *and* $\mathbf{f}$) *of depth* d is defined recursively as follows:

(1) Every member of $\mathbf{x}$ and every 0-ary function symbol (constant symbol) of $\mathbf{f}$ is a term of depth 0.

(2) If f is a k-ary function symbol in $\mathbf{f}$ with $k \geq 1$, and if $t_1, \ldots, t_k$ are terms, with maximum depth $d - 1$, then $f(t_1, \ldots, t_k)$ is a term of depth d.

THEOREM 5.6. *Let σ be a second-order tgd. Then there is a polynomial p, which depends only on σ, with the following property. If $\langle I, J\rangle$ is an instance with active domain of size N, and if U and U' are sets (finite or infinite) that each contain the active domain and are of size at least $p(N)$, then $\langle U; I, J\rangle \models \sigma$ if and only $\langle U'; I, J\rangle \models \sigma$.*

PROOF. Let $\mathbf{f}$ be the collection of function symbols that appear in σ, and let $\mathbf{x}$ be a collection of variables. It is straightforward to verify that for each d, there is a polynomial p_d with nonnegative coefficients, where p_d depends only on $\mathbf{f}$, such that the number of terms based on $\mathbf{x}$ and $\mathbf{f}$, of depth at most d, is at most $p_d(m)$, where m is the size of $\mathbf{x}$.

Let $\langle I, J\rangle$ be an instance with active domain D of size N. Let D_I, of size N_I, be the active domain of I. We refer to the set of terms based on D_I and $\mathbf{f}$ as the *Herbrand universe*. For each s, let H_s denote the set of members of the Herbrand universe of depth at most s.

Let σ be the SO tgd (3). Let U and U' be sets (finite or infinite) that each contain the active domain and are of size at least $p_d(N_I)$. We shall show that $\langle U; I, J\rangle \models \sigma$ if and only $\langle U'; I, J\rangle \models \sigma$. This is sufficient to prove the theorem, since the facts that $N_I \leq N$ and that p_d has nonnegative coefficients immediately imply that $p_d(N_I) \leq p_d(N)$. Intuitively, the key to the proof is that σ refers only to members of H_d. Assume without loss of generality that the size $|U'|$ of U' is at most the size $|U|$ of U. By renaming members of U if necessary, we can assume that $U' \subseteq U$.

Assume first that $\langle U'; I, J\rangle \models \sigma$; we shall show that $\langle U; I, J\rangle \models \sigma$. Since $\langle U'; I, J\rangle \models \sigma$, there is a collection $\mathbf{f'^0}$ of functions with domain and range U' such that whenever $1 \leq i \leq n$ and $\mathbf{x_i} \mapsto \mathbf{a_i}$ is an assignment of $\mathbf{x_i}$ to members of D_I, we have $\langle U'; I, J\rangle \models (\phi_i \rightarrow \psi_i)[\mathbf{f} \mapsto \mathbf{f'^0}, \mathbf{x_i} \mapsto \mathbf{a_i}]$. Extend every function f'^0 in $\mathbf{f'^0}$ to a function f^0 with domain and range U by letting $f^0(a) = f'^0(a)$ for $a \in U'$ and letting $f^0(a)$ be an arbitrary member of U otherwise. Let $\mathbf{f^0}$ be the collection of these extensions f^0. It is easy to see that the interpretation of the members of the Herbrand universe (of arbitrary depth), under the assignment $\mathbf{f} \mapsto \mathbf{f^0}$, lies in U', since each f^0 maps U' into U'. Since every term in $\phi_i \rightarrow \psi_i$ refers to members of the Herbrand universe, and since $\mathbf{f^0}$ and $\mathbf{f'^0}$ agree on U'. it follows that for $1 \leq i \leq n$, we have $\langle U; I, J\rangle \models (\phi_i \rightarrow \psi_i)[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto \mathbf{a_i}]$. Therefore, $\langle U; I, J\rangle \models \sigma$, which was to be shown.

Conversely, assume that $\langle U; I, J\rangle \models \sigma$; we shall show that $\langle U'; I, J\rangle \models \sigma$. Since $\langle U; I, J\rangle \models \sigma$, there is a collection $\mathbf{f^0}$ of functions with domain and range U such that whenever $1 \leq i \leq n$ and $\mathbf{x_i} \mapsto \mathbf{a_i}$ is an assignment of $\mathbf{x_i}$ to members of D_I, we have $\langle U; I, J\rangle \models (\phi_i \rightarrow \psi_i)[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto \mathbf{a_i}]$. For each f^0 in $\mathbf{f^0}$, define the function f'^0 (with domain and range U') so that the interpretation of members of H_d is the same using either $\mathbf{f^0}$ or $\mathbf{f'^0}$ (the size of U' is big enough that this is possible). Since every term in $\phi_i \rightarrow \psi_i$ refers only to members of H_d, it follows that for $1 \leq i \leq n$, we have $\langle U', I, J\rangle \models (\phi_i \rightarrow \psi_i)[\mathbf{f} \mapsto \mathbf{f'^0}, \mathbf{x_i} \mapsto \mathbf{a_i}]$. Therefore, $\langle U'; I, J\rangle \models \sigma$, which was to be shown. □

5.3 Model-Checking for Second-Order TGDs

We now show that model checking for second-order tgds, that is, verifying whether a pair of source and target instances satisfies a second-order tgd, is in NP and can be NP-complete. This is in contrast with the case of (first-order) source-to-target tgds, for which model checking is always in polynomial time.

THEOREM 5.7. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \sigma)$ be a schema mapping, where σ is an SO tgd. The problem of deciding, given I and J, whether $\langle I, J\rangle$ satisfies σ, is in* NP *and can be* NP*-complete.*

PROOF. The NP upper bound follows immediately from the "easy direction" of Fagin's Theorem [Fagin 1974], along with the fact (Theorem 5.6) that the universe can be taken to be of polynomial size. We now prove the lower bound.

Let the source schema $\mathbf{S}$ consist of a single binary relation symbol E, and let the target schema $\mathbf{T}$ consist of a single binary relation symbol D. Let σ be the SO tgd $\exists f(E(x, y) \rightarrow D(f(x), (f(y)))$. We now show that the problem of deciding if $\langle I, J\rangle \models \sigma$ is NP-complete.

Let J be the same as the instance I_3 in the proof of Theorem 4.6. Thus, J is the target instance with

$$D^J = \{(r, g), (g, r), (b, r), (r, b), (g, b), (b, g)\}.$$

In words, D^J contains all pairs of different colors among the three colors r, g, and b. Let $\mathbf{G} = (V, E)$ be a graph and let I be the instance over $\mathbf{S}_1$ consisting of the edge relation E of $\mathbf{G}$. We claim that $\mathbf{G}$ is 3-colorable if and only if $\langle I, J\rangle \models \sigma$. This is sufficient to prove the theorem, since 3-COLORABILITY is NP-complete.

Assume first that $\mathbf{G}$ is 3-colorable. Then there is a coloring function c that maps members of V to the set $\{r, b, g\}$ such that $c(x) \neq c(y)$ for every edge $(x, y) \in E$. It is easy to see that $\langle I, J\rangle \models (E(x, y) \rightarrow D(f(x), f(y)))[f \mapsto c]$. Therefore, $\langle I, J\rangle \models \sigma$.

Conversely, assume that $\langle I, J\rangle \models \sigma$. Then, there is c such that $\langle I, J\rangle \models (E(x, y) \rightarrow D(f(x), f(y)))[f \mapsto c]$. It is easy to see that c is a function that maps members of V to the set $\{r, b, g\}$ such that $c(x) \neq c(y)$ for every edge $(x, y) \in E$. Therefore, $\mathbf{G}$ is 3-colorable. □

Although model-checking for SO tgds can be NP-complete, there are practical problems involving SO tgds other than model-checking. For example, in the two important cases of data exchange and query answering, all that is needed is to materialize the result of data exchange given a source instance or to compute the answers to a target query given a source instance. We shall later show that SO tgds have polynomial-time properties for such scenarios. Furthermore, we shall also show that SO tgds compose. Thus, SO tgds form a good candidate for *the* schema mapping language.

6. CHASE AND DATA EXCHANGE WITH SECOND-ORDER TGDS

Our main motivation for studying composition of schema mappings stems from data exchange [Fagin et al. 2005a, 2005b]. A specific case of data exchange is one in which we are given a source schema, a target schema, and a schema mapping specified by a finite set of source-to-target tgds. Given an instance over the source schema, we are interested in materializing a target instance that satisfies the specification. In the case of two or more successive data exchange scenarios and when only a final instance over the final target schema is of interest, we would like to avoid materializing intermediate instances, and hence use the schema mapping that is the composition of the sequence of schema mappings. However, as we have argued so far, the language of source-to-target tgds may no longer be appropriate in this case. We instead use second-order tgds.

In Section 6.1, we modify the classical chase technique [Beeri and Vardi 1984a] to handle SO tgds (rather than the usual first-order tgds). In Section 6.2, we prove a technical lemma about chasing with SO tgds. We subsequently use this lemma in Section 6.3 to show that the chase with SO tgds yields a universal solution for data exchange (as is the case with first-order tgds [Fagin et al. 2005a]). We also show that chasing with SO tgds is a polynomial-time procedure (polynomial in the size of the source instance). As a consequence, computing the certain answers of conjunctive queries in data exchange settings specified by SO tgds can be done in polynomial time.

6.1 The Chase with Second-Order TGDs

We first define ground terms and ground instances. Given a set $\mathbf{V}$ of values and a collection $\mathbf{f}$ of function symbols, *a ground term u over $\mathbf{V}$ and $\mathbf{f}$* is defined recursively as either a value of $\mathbf{V}$ or a function term of the form $f(u_1, \ldots, u_k)$, where $u_1, \ldots, u_k$ are ground terms over $\mathbf{V}$ and $\mathbf{f}$, and f is a k-ary function symbol in $\mathbf{f}$. *A ground instance, with respect to* $\mathbf{V}$ *and* $\mathbf{f}$, is an instance whose values are ground terms over $\mathbf{V}$ and $\mathbf{f}$. Note that an instance (in the usual sense) with values in $\mathbf{V}$ is also a ground instance over $\mathbf{V}$ and $\mathbf{f}$ (for every $\mathbf{f}$). Homomorphisms are defined on ground instances in the same way they are defined on usual instances. The only difference is that the domain of such homomorphisms includes now all the ground terms over $\mathbf{V}$ and $\mathbf{f}$.

Using an example, we illustrate next, informally, the chase with SO tgds.

Example 6.1. Consider the schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st})$ where Σ_{st} is the following SO tgd:

$$\begin{aligned}&\exists f (\forall x \forall y\, (R(x, y) \to U(x, y, f(x)))\\ &\qquad \wedge\ \forall x \forall x' \forall y \forall y' (R(x, y) \wedge R(x', y') \wedge (f(x) = f(x')) \to T(y, y'))).\end{aligned}$$

Each of the formulas of the form $\forall \mathbf{x}(\phi_1 \to \phi_2)$ that appear under the scope of the existentially quantified functions can be thought of as a source-to-target "tgd". The difference from a normal source-to-target tgd is that now we can have function terms in the relational atoms of ϕ_2, as well as equality atoms in ϕ_1, and we do not have any existential quantifiers in ϕ_2. Suppose now that we are given a source instance I where R consists of the following three tuples: (a, b), (a, c), and (d, e). The chase starts with an instance of the form $\langle I, \emptyset \rangle$ and constructs an instance of the form $\langle I, J \rangle$ by applying all the "tgds" until these "tgds" are all satisfied. A "tgd" is applied when the left-hand side ϕ of the "tgd" can be mapped to I but the corresponding right-hand side ψ does not yet exist in J, in which case we add it to J. By applying the first "tgd" in Σ_{st}, for the first tuple (a, b) of R we generate a tuple $(a, b, f(a))$ in U. In applying the same "tgd", this time for the tuple (a, c) of R, we generate $(a, c, f(a))$ in U (the same ground function term $f(a)$ appears again). Finally, for the last tuple (d, e) and the same "tgd" we generate $(d, e, f(d))$ in U. Note that the values that may now appear in tuples of J are ground terms over the set of source values of I and over the singleton set $\{f\}$ of function symbols.

To apply the second "tgd" in Σ_{st}, we see that only the combinations $f(a) = f(a)$ and $f(d) = f(d)$ can satisfy the equality $f(x) = f(x')$. (Two ground terms are treated as equal precisely if they are syntactically identical.) Hence, the chase will generate the tuples (b, b), (b, c), (c, b), (c, c) and (e, e) in T.

At the end of the chase, the resulting instance satisfies all the "tgds". This instance is formed with source values together with ground function terms that are added during the chase.

Note that we view each one of the ground function terms as a distinct value. In practice, one can substitute the ground function terms with values from a concrete domain such that the term structure is "forgotten". For example, we could replace all occurrences of the term $f(a)$ in J with a null X. However, such replacement is more of an implementation issue that is orthogonal to the general concepts and results that we will give here.

We now give the formal details of the chase with second-order tgds. In the following definitions, whenever we refer to a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \sigma)$

where σ is a second-order tgd, we assume that σ has the general form:

$$\exists \mathbf{f}((\forall \mathbf{x_1}(\phi_1 \rightarrow \psi_1)) \wedge \cdots \wedge (\forall \mathbf{x_n}(\phi_n \rightarrow \psi_n))).$$

For each i, we may, as before, denote the conjunct $\forall \mathbf{x_i}(\phi_i \rightarrow \psi_i)$ of σ by C_i. We start by defining the notion of homomorphism from an SO tgd to an instance. We first introduce an auxiliary notion.

Definition 6.2. Let $\mathcal{M}$ be a schema mapping defined by a second-order tgd. Let I be a source instance and let h be a function mapping the variables in $\mathbf{x_i}$ into values of I. Let t and t' be terms over $\mathbf{x_i}$ and $\mathbf{f}$. We say that *the equality $t = t'$ is satisfied in I under h* if: (1) the equality is of the form $x = x'$, and $h(x)$ and $h(x')$ are the same value, or (2) the equality is of the form $f(t_1, \ldots, t_l) = f(t'_1, \ldots, t'_l)$ where f is in $\mathbf{f}$, and the equalities $t_1 = t'_1, \ldots, t_l = t'_l$ are satisfied in I under h. (Note that the definition is recursive.)

Definition 6.3. Let $\mathcal{M}$ be a schema mapping defined by a second-order tgd. Let I be a source instance and let h be a function mapping the variables in $\mathbf{x_i}$ into values of I. We say that h is a *homomorphism from the conjunct C_i of σ to the instance I* if the following conditions hold: (1) for every relational atom $S(y_1, \ldots, y_k)$ in ϕ_i, the tuple $(h(y_1), \ldots, h(y_k))$ is in S^I, and (2) every equality in ϕ_i is satisfied in I under h. In the literature, what we call in this context a homomorphism is sometimes called a *valuation*, or a *variable assignment* [Abiteboul et al. 1995].

We extend h on terms in the natural way by defining $h(f(t_1, \ldots, t_l))$ to be $f(h(t_1), \ldots, h(t_l))$ for every term $f(t_1, \ldots, t_l)$ that occurs in ψ_i.

Definition 6.4 (*Chase Step*). Let $\mathcal{M}$ be a schema mapping defined by a second-order tgd σ. Let $\mathbf{V}$ be a set of values and let I be an instance, with values in $\mathbf{V}$, over the source schema $\mathbf{S}$. Furthermore, let J_1 be a ground instance, with respect to $\mathbf{V}$ and $\mathbf{f}$, over the target schema $\mathbf{T}$.

Assume that there is a homomorphism h from some conjunct $C_i = \forall \mathbf{x}(\phi_i \rightarrow \psi_i)$ of σ into I with the property there is at least one atomic formula $T(t_1, \ldots, t_p)$ in ψ_i such that $(h(t_1), \ldots, h(t_p))$ is not a tuple in T^{J_1}. We say that *C_i can be applied to $\langle I, J_1\rangle$ with homomorphism h.*

Furthermore, let J_2 be the ground instance with respect to $\mathbf{V}$ and $\mathbf{f}$ that is defined as follows: for every target relation T, let T^{J_2} be the union of T^{J_1} with the set of all tuples $(h(t_1), \ldots, h(t_p))$ where $T(t_1, \ldots, t_p)$ is an atomic formula in ψ_i. We say that $\langle I, J_2\rangle$ is *the result of applying C_i to $\langle I, J_1\rangle$ with h* and write $\langle I, J_1\rangle \xrightarrow{C_i,h} \langle I, J_2\rangle$. We also call this a *chase step*.

In the following, as before, we will denote by $\emptyset$ an empty instance.

Definition 6.5 (*Chase*). Let $\mathcal{M}$ be a schema mapping where σ is a second-order tgd, and let I be a source instance.

(1) A *chase sequence of $\langle I, \emptyset\rangle$ with σ* is a finite sequence of chase steps $\langle I, J_k\rangle \xrightarrow{C_k,h_k} \langle I, J_{k+1}\rangle$, for $0 \leq k < m$, with $J_0 = \emptyset$ and C_k a conjunct of σ.

(2) A *chase of $\langle I, \emptyset\rangle$ with σ* is a chase sequence $\langle I, J_k\rangle \xrightarrow{C_k,h_k} \langle I, J_{k+1}\rangle$, for $0 \leq k < m$, such that it is not the case that there is a conjunct C_i of σ and a

homomorphism h where C_i can be applied to $\langle I, J_m \rangle$ with h. We say that $\langle I, J_m \rangle$ is the result of this chase.

It is easy to verify that if $\langle I, J \rangle$ is the result of some chase with a second-order tgd σ then $\langle I, J \rangle$ satisfies σ. Indeed, we can take the universe U to be the set of all the ground terms over $\mathbf{V}$ and $\mathbf{f}$. (This universe includes the active domains of I and J.) Then, $\langle U; I, J \rangle$ satisfies σ. In particular, for each f in $\mathbf{f}$, we take the following interpretation: assuming that f is k-ary, we define the value of f applied to $u_1, \ldots, u_k$ (where $u_1, \ldots, u_k$ are ground terms over $\mathbf{V}$ and $\mathbf{f}$) to be precisely the ground term $f(u_1, \ldots, u_k)$. It can be seen that under this interpretation for $\mathbf{f}$, we have that $\langle U; I, J \rangle$ satisfies all the conjuncts of σ (otherwise, additional chase steps would be applicable). Moreover, Theorem 5.6 says that we can change U to an arbitrary universe U' that includes the active domain of I and J and is sufficiently large, and we still have that $\langle U'; I, J \rangle$ satisfies σ.

We now make the following observation to compare and contrast chasing with SO tgds and chasing with first-order tgds. Although complicated by the presence of function symbols and equalities, chasing with SO tgds is at the same time simpler than chasing with first-order tgds due to the following. There is an explicit partitioning of the schema into the source schema $\mathbf{S}$ and the target schema $\mathbf{T}$; moreover, the source instance I is never changed during the chase and the homomorphisms from conjuncts of σ that can apply during the chase are all homomorphisms into I. Hence, it is possible to enumerate a priori, before the chase, all the homomorphisms that will ever apply during the chase (since they do not depend on J). In fact, the number of homomorphisms can be precisely bounded to be polynomial in the size of the given source instance. Consequently, the chase with second-order tgds takes time polynomial in the size of I. We will give a precise analysis in Section 6.3.

The above observation can also be used to give an equivalent, more declarative, formulation of the chase with SO tgds.[3] We make this precise by the following proposition, which is an immediate consequence of the fact that all homomorphisms that ever apply during the chase can be enumerated before the chase.

PROPOSITION 6.6. *Let σ be a second-order tgd and let I be an instance over* $\mathbf{S}$. *For each conjunct C of σ and each homomorphism h from C into I, let $J_{C,h}$ be the ground instance that contains a tuple $(h(t_1), \ldots, h(t_p))$ in $T^{J_{C,h}}$ whenever there is an atomic formula $T(t_1, \ldots, t_p)$ in the right-hand side of C. Then, the following are equivalent:*

(1) *$\langle I, J_m \rangle$ is the result of a chase of $\langle I, \emptyset \rangle$.*

(2) *J_m consists of the union (relation by relation) of all $J_{C,h}$ over all conjuncts C of σ and all homomorphisms h from C into I.*

The above proposition also shows that for every two chases of $\langle I, \emptyset \rangle$ with a second-order tgd σ, with results $\langle I, J \rangle$ and, respectively, $\langle I, J' \rangle$, it is the case

[3] This formulation was suggested by one of the referees of this article.

that J and J' are identical (since they are both equal to the union of instances stated in (2)). In other words, chasing with second-order tgds is Church-Rosser.

6.2 A Basic Property of the Chase with Second-Order TGDs

We next prove a technical lemma about chasing with second-order tgds. This lemma, that we shall subsequently use, is a variation of a known result in the case of chasing with (first-order) tgds [Beeri and Vardi 1984a; Fagin et al. 2005a].

LEMMA 6.7. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \sigma)$ be a schema mapping where σ is a second-order tgd of the form:*

$$\exists \mathbf{f}((\forall \mathbf{x_1}(\phi_1 \rightarrow \psi_1)) \wedge \cdots \wedge (\forall \mathbf{x_n}(\phi_n \rightarrow \psi_n))).$$

Let $\langle I', J'\rangle$ be an instance over the schema $\langle \mathbf{S}, \mathbf{T}\rangle$ such that $\langle I', J'\rangle$ satisfies σ, that is, $\langle U'; I', J'\rangle$ satisfies σ, for a countably infinite universe U' that includes the active domain. Moreover, let $\mathbf{f^0}$ be a collection of functions over U' such that

$$\langle I', J'\rangle \models \bigwedge_i (\forall \mathbf{x_i}(\phi_i \rightarrow \psi_i))[\mathbf{f} \mapsto \mathbf{f^0}].$$

Let I be an instance over $\mathbf{S}$, with values in some domain $\mathbf{V}$, and let J_1 and J_2 be two ground instances with respect to $\mathbf{V}$ and $\mathbf{f}$ such that $\langle I, J_1\rangle \xrightarrow{C_i,h} \langle I, J_2\rangle$ is a chase step with some conjunct C_i of σ and some homomorphism h. Assume that g is a homomorphism from $\langle I, J_1\rangle$ to $\langle I', J'\rangle$ such that:

$$(*) \qquad g(f(u_1, \ldots, u_k)) = f^0(g(u_1), \ldots, g(u_k)),$$

for every ground function term $f(u_1, \ldots, u_k)$ over $\mathbf{V}$ and $\mathbf{f}$.

Then, g is a homomorphism from $\langle I, J_2\rangle$ to $\langle I', J'\rangle$.

PROOF. We first show that the function $g \circ h$ from the variables $\mathbf{x_i}$ of C_i to U' satisfies the following two properties:

(1) for every atom $S(y_1, \ldots, y_k)$ in ϕ_i (where we recall that ϕ_i denotes the left-hand side of the implication in C_i), the tuple $(g \circ h(y_1), \ldots, g \circ h(y_k))$ is in $S^{I'}$, and

(2) for every equality $t = t'$ in ϕ_i, we have that:

$$(**) \qquad t[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})] = t'[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})]$$

(i.e., the two members of the equation represent the same value of U').

We verify (1) first. Since h is a homomorphism from the conjunct C_i into I, it follows by Definition 6.3 that the tuple $(h(y_1), \ldots, h(y_k))$ is in S^I. Moreover, g is a homomorphism from I to I'. Hence, the tuple $(g(h(y_1)), \ldots, g(h(y_k)))$ is in $S^{I'}$.

As for property (2), we prove the following stronger statement: For every equality $t = t'$ that is satisfied in I under h, we have that condition (**) holds. Since h is a homomorphism from C_i into I, it must be the case that every equality of ϕ_i is satisfied in I under h. Therefore, property (2) is proven under the assumption that the stronger statement holds.

The proof of the stronger statement is by induction on the structure of $t = t'$.

Base Case: The equality $t = t'$ is of the form $x = x'$. Then it must be the case that $h(x) = h(x')$ and therefore $g \circ h(x) = g \circ h(x')$. This is the same as saying that (**) holds for $x = x'$.

Inductive Case: The equality $t = t'$ is of the form $f(t_1, \ldots, t_l) = f(t'_1, \ldots, t'_l)$ where f is in $\mathbf{f}$ and the equalities $t_1 = t'_1, \ldots, t_l = t'_l$ are satisfied in I under h. By the inductive hypothesis, we have that:

$$t_1[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})] = t'_1[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})]$$
$$\ldots$$
$$t_l[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})] = t'_l[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})].$$

It then follows that the following equality holds:

$$f^{\,0}(t_1[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})], \ldots, t_l[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})])$$
$$=$$
$$f^{\,0}(t'_1[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})], \ldots, t'_l[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})]).$$

But this equality is the same as saying that condition (**) holds for $f(t_1, \ldots, t_l) = f(t'_1, \ldots, t'_l)$. This concludes the proof of the inductive case.

So far, we have shown that $g \circ h$ is an assignment for the variables in $\mathbf{x_i}$ with values in U' such that properties (1) and (2) hold. This is the same as saying that $I' \models \phi_i[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})]$. Since $\langle I', J' \rangle \models (\forall \mathbf{x_i}(\phi_i \rightarrow \psi_i))[\mathbf{f} \mapsto \mathbf{f^0}]$ (where we recall that ψ_i denotes the right-hand side of the implication in C_i), it must be the case that $J' \models \psi_i[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})]$. In other words, for every atom $T(t_1, \ldots, t_p)$ of ψ_i, the tuple $\mathbf{u}$, which is defined as

$$(t_1[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})], \ldots, t_p[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})]),$$

is in $T^{J'}$. But as we now show, this tuple $\mathbf{u}$ is the same as the tuple $(g \circ h(t_1), \ldots, g \circ h(t_p))$. First, for $1 \leq l \leq p$, we have that $h(t_l) = t_l[\mathbf{x_i} \mapsto h(\mathbf{x_i})]$, from the way h is defined on terms (see Definition 6.3). It follows that $g(h(t_l)) = g(t_l[\mathbf{x_i} \mapsto h(\mathbf{x_i})]$, for $1 \leq l \leq p$. Since g satisfies condition (*), it is the case that

$$g(t_l[\mathbf{x_i} \mapsto h(\mathbf{x_i})]) = t_l[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})],$$

for $1 \leq l \leq p$. Hence, $g \circ h(t_l) = g(h(t_l)) = t_l[\mathbf{f} \mapsto \mathbf{f^0}, \mathbf{x_i} \mapsto g \circ h(\mathbf{x_i})]$, for $1 \leq l \leq p$. In other words, the tuple $(g \circ h(t_1), \ldots, g \circ h(t_p))$ is the same as $\mathbf{u}$. Putting this together with the earlier fact that for every atom $T(t_1, \ldots, t_p)$ of ψ_i, the tuple $\mathbf{u}$ is in $T^{J'}$, we obtain that for every atom $T(t_1, \ldots, t_p)$ of ψ_i, the tuple $(g \circ h(t_1), \ldots, g \circ h(t_p))$ is in $T^{J'}$.

We can show now that g is a homomorphism from $\langle I, J_2 \rangle$ to $\langle I', J' \rangle$. It is enough to show that the image, under g, of each of the "new " tuples in J_2, that are added during the chase step with C_i and h, is a tuple in the corresponding relation of J'. Indeed, let $(h(t_1), \ldots, h(t_p))$ be a "new" tuple of T^{J_2}, for some atom $T(t_1, \ldots, t_p)$ of ψ_i. We need to prove that $(g(h(t_1)), \ldots, g(h(t_p)))$ is in $T^{J'}$. But we have just shown this, for every atom $T(t_1, \ldots, t_p)$ of ψ_i. This concludes the proof. □

6.3 Data Exchange and Query Answering with Second-Order TGDs

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st})$ be a schema mapping where Σ_{st} contains only source-to-target tgds, and let I be a source instance over the schema $\mathbf{S}$. It is known [Fagin et al. 2005a] that chasing I with Σ_{st} produces, in polynomial time in the size of I, a universal solution for I under $\mathcal{M}$. The next theorem asserts that a similar result holds when we chase a source instance I with SO tgds.

THEOREM 6.8. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \sigma)$ be a schema mapping where σ is an SO tgd. Then for every source instance I over $\mathbf{S}$, chasing $\langle I, \emptyset\rangle$ with σ terminates in polynomial time (in the size of I) with a result $\langle I, J\rangle$. Moreover, J is a universal solution for I under $\mathcal{M}$.*

PROOF. We first prove that the chase terminates in time polynomial in the size of I. We consider $\mathcal{M}$ fixed. Let k be the maximum number of universally quantified variables in a conjunct of σ, let n be the total number of distinct values in I, and let c be the total number of conjuncts in σ. For a given conjunct C of σ, there can be at most n^k homomorphisms. Since there are c conjuncts, the total number of homomorphisms from σ into I is at most $c \times n^k$. Each such homomorphism can yield at most one chase step of I with σ. (Once a chase step with a homomorphism h from a conjunct C is applied, then there cannot be another chase step with the same homomorphism and same conjunct, because all the "required" target tuples have already been added in the first chase step.) Furthermore, I is not modified by the chase; hence, no new homomorphisms can arise during the chase. Therefore, we can have at most $c \times n^k$ chase steps.

We now estimate the time spent during one chase step. Let t be the maximum number of atoms in ψ, over all conjuncts $\forall\mathbf{x}(\phi \rightarrow \psi)$ in σ. Let m be the total number of tuples that will exist in the target after the chase. This number m is bounded by the number of chase steps, which is $c \times n^k$, times the number of tuples that can be added in one chase step, which is at most t. Thus, m is at most $t \times c \times n^k$.

At each chase step, we may spend $c \times n^k \times q(n)$ time to search for an applicable homomorphism. Here, $c \times n^k$ is the maximum number of functions that we may need to search through, while $q(n)$ is the time to check whether such a function is a homomorphism or not, using Definition 6.3. It is easy to verify that q is a polynomial in n. (Note that we could actually reduce the above time $c \times n^k \times q(n)$, if we enumerate the list of all the candidate homomorphisms before the chase, and then at each chase step we pick the next homomorphism from this list. This is an optimization that does not affect the overall upper bound.) In addition, if the chase step involves a conjunct $\forall\mathbf{x}(\phi \rightarrow \psi)$, for each of the atoms in ψ we need to check whether the corresponding tuple already exists in the target, before adding it into the target. This takes at most t (the maximum number of atoms in ψ) times m (the maximum number of tuples in the target), or at most $t^2 \times c \times n^k$. Thus, the time spent at each chase step is at most $t^2 \times c \times n^k + c \times n^k \times q(n)$.

Overall, the time to chase is at most $c \times n^k$, the number of chase steps, times $t^2 \times c \times n^k + c \times n^k \times q(n)$, the time spent at each chase step. This number is a polynomial in the size of I.

We now prove that J is a universal solution for I under $\mathcal{M}$. We will make use of Lemma 6.7. Assume that σ is of the form:

$$\exists \mathbf{f}((\forall \mathbf{x_1}(\phi_1 \rightarrow \psi_1)) \wedge \cdots \wedge (\forall \mathbf{x_n}(\phi_n \rightarrow \psi_n))).$$

Let K be an arbitrary solution for I under $\mathcal{M}$. Thus, $\langle U; I, K\rangle \models \sigma$, where U is the universe. Let $\mathbf{f^0}$ be a collection of functions over U such that

$$\langle I, K\rangle \models \bigwedge_i (\forall \mathbf{x_i}(\phi_i \rightarrow \psi_i))[\mathbf{f} \mapsto \mathbf{f^0}].$$

Let us denote with $\mathbf{V}$ the set of values in I, and let g be the identity function on $\mathbf{V}$. We extend g to ground terms over $\mathbf{V}$ and $\mathbf{f}$, by defining $g'(v) = g(v)$, for every value v in $\mathbf{V}$, and $g'(f(u_1, \ldots, u_k)) = f^0(g'(u_1), \ldots, g'(u_k))$, for every ground function term $f(u_1, \ldots, u_k)$ over $\mathbf{V}$ and $\mathbf{f}$. It is immediate that g' is a homomorphism from $\langle I, \emptyset\rangle$ to $\langle I, K\rangle$ (since g is a homomorphism from I to I and there are no tuples in the target side of $\langle I, \emptyset\rangle$).

By definition of g', we have that g' satisfies the condition (*) from Lemma 6.7, where g' plays the role of g. Hence the lemma becomes applicable at every chase step in the chase sequence from $\langle I, \emptyset\rangle$ to $\langle I, J\rangle$. We obtain that $g' : \langle I, J\rangle \rightarrow \langle I, K\rangle$ is a homomorphism. In particular, g' is a homomorphism from J to K satisfying $g'(v) = g(v)$ whenever v is in $\mathbf{V}$. Since g is the identity function on $\mathbf{V}$, we obtain that g' is a homomorphism from J to K satisfying $g'(v) = v$ whenever v is in $\mathbf{V}$. Since K was picked to be an arbitrary solution of I under $\mathcal{M}$, we conclude that J is a universal solution for I under $\mathcal{M}$. □

The above theorem has an immediate but important consequence in terms of query answering over the target schema. Let us recall the definitions of conjunctive queries (with and without inequalities, since we will make use of conjunctive queries with inequalities later), and unions of conjunctive queries. A *conjunctive query* $q(\mathbf{x})$ is a formula of the form $\exists \mathbf{y} \phi(\mathbf{x}, \mathbf{y})$ where $\phi(\mathbf{x}, \mathbf{y})$ is a conjunction of atomic formulas. If, in addition to atomic formulas, the conjunction $\phi(\mathbf{x}, \mathbf{y})$ is allowed to contain inequalities of the form $z_i \neq z_j$, where z_i, z_j are variables among $\mathbf{x}$ and $\mathbf{y}$, we call $q(\mathbf{x})$ a *conjunctive query with inequalities.* We also impose a safety condition, that every variable in $\mathbf{x}$ and $\mathbf{y}$ must appear in an atomic formula, not just in an inequality. A *union of conjunctive queries* is a disjunction $q(\mathbf{x}) = q_1(\mathbf{x}) \vee \cdots \vee q_n(\mathbf{x})$ where $q_1(\mathbf{x}), \ldots, q_n(\mathbf{x})$ are conjunctive queries.

It was shown in Fagin et al. [2005a] that if J is a universal solution for I under $\mathcal{M}$ and q is a union of conjunctive queries, then $\mathit{certain}_{\mathcal{M}}(q, I)$ equals $q(J)_{\downarrow}$, which is the result of evaluating q on J and then keeping only those tuples formed entirely of values from I. The equality $\mathit{certain}_{\mathcal{M}}(q, I) = q(J)_{\downarrow}$ holds for arbitrarily specified schema mappings $\mathcal{M}$. In particular, it holds for schema mappings specified by SO tgds. This fact, taken together with Theorem 6.8, implies the following result.

COROLLARY 6.9. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \sigma)$ be a schema mapping where σ is an SO tgd. Let q be a union of conjunctive queries over the target schema $\mathbf{T}$. Then, for every source instance I over $\mathbf{S}$, the set $\mathit{certain}_{\mathcal{M}}(q, I)$ can be computed in polynomial time (in the size of I).*

We point out an interesting contrast between the above result and one of the results on query answering given in Abiteboul and Duschka [1998]. There it was shown that when the source schema is described in terms of the target schema by means of arbitrary first-order views, computing the certain answers of conjunctive queries becomes undecidable. In contrast, our result shows that although the schema mappings that we consider go beyond first-order, computing the certain answers of unions of conjunctive queries remains in polynomial time, as it is with schema mappings specified by source-to-target tgds [Fagin et al. 2005a]. Thus, second-order tgds form a well-behaved fragment of second-order logic, since for the purposes of data exchange and query answering, second-order tgds behave similarly to source-to-target tgds.

7. COMPOSABILITY OF SECOND-ORDER TGDS

As we saw in Theorem 4.5, sets of source-to-target tgds are not closed under composition. By contrast, we show that SO tgds are closed under composition. That is, given two schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ where Σ_{12} and Σ_{23} are SO tgds, the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is always definable by an SO tgd. We show this by exhibiting a composition algorithm in this section and then showing that the composition algorithm is correct.

Algorithm Compose($\mathcal{M}_{12}$, $\mathcal{M}_{23}$)

Input: Two schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where Σ_{12} and Σ_{23} are SO tgds.
Output: A schema mapping $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$, which is the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ and where Σ_{13} is an SO tgd.

1. (*Normalize the SO tgds in* Σ_{12} *and* Σ_{23}.)
   Rename the function symbols so that the function symbols that appear in Σ_{12} are all distinct from the function symbols that appear in Σ_{23}. For notational convenience, we shall refer to variables in Σ_{12} as x's, possibly with subscripts, and the variables in Σ_{23} as y's, possibly with subscripts. Initialize $\mathcal{S}_{12}$ and $\mathcal{S}_{23}$ to each be the empty set. Assume that the SO tgd in Σ_{12} is
   $$\exists \mathbf{f}((\forall \mathbf{x_1}(\phi_1 \rightarrow \psi_1)) \wedge \cdots \wedge (\forall \mathbf{x_n}(\phi_n \rightarrow \psi_n))).$$
   Put each of the n implications $\phi_i \rightarrow \psi_i$, for $1 \leq i \leq n$, into $\mathcal{S}_{12}$. We do likewise for Σ_{23} and $\mathcal{S}_{23}$. Each implication χ in $\mathcal{S}_{12}$ has the form $\phi(\mathbf{x}) \rightarrow \bigwedge_{j=1}^{k} R_j(\mathbf{t_j})$ where every member of $\mathbf{x}$ is a universally quantified variable, and each $\mathbf{t_j}$, for $1 \leq j \leq k$, is a sequence of terms over $\mathbf{x}$. We then replace each such implication χ in $\mathcal{S}_{12}$ with k implications:
   $$\phi(\mathbf{x}) \rightarrow R_1(\mathbf{t_1}), \ldots, \ \phi(\mathbf{x}) \rightarrow R_k(\mathbf{t_k}).$$
2. (*Compose* $\mathcal{S}_{12}$ *with* $\mathcal{S}_{23}$.)
   Repeat the following until every relation symbol in the left-hand side of every formula in $\mathcal{S}_{23}$ is from $\mathbf{S}_1$.
   For each implication χ in $\mathcal{S}_{23}$ of the form $\psi \rightarrow \gamma$ where there is an atom $R(\mathbf{y})$ in ψ such that R is a relation symbol in $\mathbf{S}_2$, we perform the following steps to replace $R(\mathbf{y})$ with atoms over $\mathbf{S}_1$. (The equalities in ψ are left unchanged.) Let
   $$\phi_1 \rightarrow R(\mathbf{t_1}), \ldots, \ \phi_p \rightarrow R(\mathbf{t_p})$$
   be all the implications in $\mathcal{S}_{12}$ whose right-hand side has the relation symbol R in it. If no such implications exist in $\mathcal{S}_{12}$, we remove χ from $\mathcal{S}_{23}$. Otherwise, for each such implication $\phi_i \rightarrow R(\mathbf{t_i})$, rename the variables in this implication so that they do not

overlap with the variables in χ. (In fact, every time we compose with this implication, we take a fresh copy of the implication, with new variables.) Let θ_i be the conjunction of the equalities between the variables in $R(\mathbf{y})$ and the corresponding terms in $R(\mathbf{t_i})$, position by position. For example, the conjunction of equalities, position by position, between $R(y_1, y_2, y_3)$ and $R(x_1, f_2(x_2), f_1(x_3))$ is $(y_1 = x_1) \wedge (y_2 = f_2(x_2)) \wedge (y_3 = f_1(x_3))$. Observe that every equality that is generated has the form $y = t$ where y is a variable in Σ_{23} and t is a term based on variables in Σ_{12} and on $\mathbf{f}$. Remove χ from $\mathcal{S}_{23}$ and add p implications to $\mathcal{S}_{23}$ as follows: replace $R(\mathbf{y})$ in χ with $\phi_i \wedge \theta_i$ and add the resulting implication to $\mathcal{S}_{23}$, for $1 \le i \le p$.

3. (*Remove variables originally in* Σ_{23}.)
For each implication χ constructed in the previous step, perform the following operation until every variable y from Σ_{23} is removed. Select an equality $y = t$ that was generated in the previous step (thus, y is a variable in Σ_{23}, and t is a term based on variables in Σ_{12} and on $\mathbf{f}$). Remove the equality $y = t$ from χ and replace every remaining occurrence of y in χ by t.
4. (*Construct* $\mathcal{M}_{13}$.)
Let $\mathcal{S}_{23} = \{\chi_1, \dots, \chi_r\}$ where $\chi_1, \dots, \chi_r$ are all the implications from the previous step. Let Σ_{13} be the following SO tgd:

$$\exists \mathbf{g} (\forall \mathbf{z_1} \chi_1 \wedge \cdots \wedge \forall \mathbf{z_r} \chi_r)$$

where $\mathbf{g}$ is the collection of all the function symbols that appear in any of the implications in $\mathcal{S}_{23}$, and where the variables in $\mathbf{z_i}$ are all the variables found in the implication χ_i, for $1 \le i \le r$.
Return $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$.

The need in Step 2 for taking a fresh copy of an implication $\phi_i \to R(\mathbf{t_i})$ in $\mathcal{S}_{12}$ with new variables each time we compose with it, rather than simply renaming the variables once at the beginning, arises when we compose this implication with an implication χ in $\mathcal{S}_{23}$ where the relational symbol R appears multiple times in the left-hand side of χ.

It is straightforward to verify that in Step 3, every variable y originally in Σ_{23} is indeed removed, so that the only remaining variables are among the variables x originally in Σ_{12}. This is because every variable y originally in Σ_{23} that is in an implication χ that remains after Step 2 appears in an equality $y = t$ that is introduced in Step 2. Then y is removed in Step 3. It follows easily that the safety condition of Definition 5.3 continues to hold, and so the algorithm generates second-order tgds that are valid according to Definition 5.3.

Note that the number of formulas in the set $\mathcal{S}_{13}$, and hence the size of Σ_{13}, is exponential in the maximum number of relational atoms that can appear in the left-hand side of an implication in Σ_{23}. In Section 7.2, we give an exponential lower bound, which shows that this exponentiality is unavoidable.

We can make use of the algorithm to compose schema mappings where Σ_{12} and Σ_{23} are specified by finite sets of source-to-target tgds by first transforming each of Σ_{12} and Σ_{23} into an SO tgd (by using the Skolemization described in Section 5) and then passing the resulting schema mappings to the composition algorithm.

Example 7.1. We illustrate the steps of the composition algorithm using the schema mappings of Example 5.2. We transform Σ_{12} and Σ_{23} into the following

SO tgds, Σ'_{12} and Σ'_{23}:

$$\Sigma'_{12} : \exists f(\forall e(\texttt{Emp}(e) \rightarrow \texttt{Mgr}_1(e, f(e)))) \qquad \begin{aligned}\Sigma'_{23} : \ & \forall e \forall m(\texttt{Mgr}_1(e, m) \rightarrow \texttt{Mgr}(e, m)) \wedge \\ & \forall e(\texttt{Mgr}_1(e, e) \rightarrow \texttt{SelfMgr}(e))\end{aligned}$$

We run the composition algorithm with $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma'_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma'_{23})$. After Step 1, the sets $\mathcal{S}_{12}$ and $\mathcal{S}_{23}$ consist of the following implications:

$$\mathcal{S}_{12} : \texttt{Emp}(e) \rightarrow \texttt{Mgr}_1(e, f(e)) \qquad \begin{aligned}\mathcal{S}_{23} : \ & \texttt{Mgr}_1(e, m) \rightarrow \texttt{Mgr}(e, m) \\ & \texttt{Mgr}_1(e, e) \rightarrow \texttt{SelfMgr}(e)\end{aligned}$$

In Step 2, we first replace the $\texttt{Mgr}_1$ atom of the first implication χ in $\mathcal{S}_{23}$ by using the implication in $\mathcal{S}_{12}$. The variable e of the implication in $\mathcal{S}_{12}$ is renamed to e_0 so that it does not overlap with the variables of χ. The result of this replacement is:

$$\texttt{Emp}(e_0) \wedge (e = e_0) \wedge (m = f(e_0)) \rightarrow \texttt{Mgr}(e, m).$$

Next, we replace the $\texttt{Mgr}_1$ atom of the second implication by using the implication in $\mathcal{S}_{12}$. The variable e of the implication in $\mathcal{S}_{12}$ is renamed to e_1 before the replacement occurs. So after Step 2, the set $\mathcal{S}_{23}$ contains two implications, as follows:

$$\texttt{Emp}(e_0) \wedge (e = e_0) \wedge (m = f(e_0)) \rightarrow \texttt{Mgr}(e, m)$$
$$\texttt{Emp}(e_1) \wedge (e = e_1) \wedge (e = f(e_1)) \rightarrow \texttt{SelfMgr}(e).$$

In Step 3, if we first remove the variable e in both implications, we are left with the following two implications:

$$\texttt{Emp}(e_0) \wedge (m = f(e_0)) \rightarrow \texttt{Mgr}(e_0, m)$$
$$\texttt{Emp}(e_1) \wedge (e_1 = f(e_1)) \rightarrow \texttt{SelfMgr}(e_1).$$

In Step 3, if we then remove the variable m from the first implication, we are left with the following two implications, which we denote by χ_1 and χ_2:

$$\begin{aligned}\chi_1 : & \quad \texttt{Emp}(e_0) \rightarrow \texttt{Mgr}(e_0, f(e_0)) \\ \chi_2 : & \quad \texttt{Emp}(e_1) \wedge (e_1 = f(e_1)) \rightarrow \texttt{SelfMgr}(e_1).\end{aligned}$$

Therefore, after Step 4, the algorithm returns $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ where Σ_{13} is the following SO tgd:

$$\exists f(\forall e_0\, \chi_1 \wedge \forall e_1\, \chi_2).$$

After substituting for χ_1 and χ_2, we obtain exactly the SO tgd that was shown in Example 5.2 (except with the variables renamed).

We note that in this example (unlike the example described immediately after the composition algorithm), it was not really necessary to rename the variable e of the implication in $\mathcal{S}_{12}$ once as e_0, and another time as e_1: it could simply have been renamed to e_0 at the beginning and not be renamed again.

7.1 Correctness of the Composition Algorithm

We now show that the above composition algorithm is correct; that is, given two schema mappings specified by second-order tgds, the algorithm returns a second-order tgd that is indeed their composition. This completely proves our earlier statement that second-order tgds are closed under composition. The correctness proof uses the chase with second-order tgds introduced in Section 6.1.

THEOREM 7.2. *Let $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where Σ_{12} and Σ_{23} are SO tgds. Then the algorithm Compose($\mathcal{M}_{12}$, $\mathcal{M}_{23}$) returns a schema mapping $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ such that Σ_{13} is an SO tgd and $\mathcal{M}_{13} = \mathcal{M}_{12} \circ \mathcal{M}_{23}$.*

PROOF. Instead of working with Σ_{13}, it is convenient to work with a slightly different but logically equivalent version Σ^*_{13}. We obtain Σ^*_{13} from Σ_{12} and Σ_{23} by making two changes to the composition algorithm. The first change is to eliminate Step 3. The second change is to modify Step 4 of the algorithm by letting $\mathbf{g}$ be the collection of all the function symbols that appear in Σ_{12} or Σ_{23}. Thus, some of these existentialized function symbols may not appear in the body (the first-order part) of Σ^*_{13}. It is easy to verify that Σ^*_{13} is logically equivalent to Σ_{13}. Because of the elimination of Step 3 of the composition algorithm, Σ^*_{13} is not necessarily an SO tgd (it may violate the safety condition). However, it is more convenient for us to prove that Σ^*_{13} defines the composition than to prove directly that Σ_{13} defines the composition. Since Σ^*_{13} is logically equivalent to Σ_{13}, this is sufficient to prove our desired result that Σ_{13} defines the composition.[4]

Let T_{12} be the body of Σ_{12}. So Σ_{12} is $\exists \mathbf{f} T_{12}$, where $\mathbf{f}$ consists of the function symbols that appear in Σ_{12}. Similarly, let T_{23} be the body of Σ_{23}, and let T^*_{13} be the body of Σ^*_{13}. In order to simplify notation, we assume (without loss of generality) that each conjunct in Σ_{12} is of the form $\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow R(\mathbf{t}))$ for a relation symbol R. If Σ_{12} is not of this form, then as we noted earlier, it can be equivalently rewritten in this form (and this is essentially what Step 1 in the algorithm does for Σ_{12}).

To show that the schema mapping $\mathcal{M}_{13}$ generated by the algorithm satisfies $\mathcal{M}_{13} = \mathcal{M}_{12} \circ \mathcal{M}_{23}$, we need to show that for every I_1 over schema $\mathbf{S_1}$ and for every I_3 over schema $\mathbf{S_3}$, we have that $\langle I_1, I_3 \rangle \models \Sigma^*_{13}$ if and only if there is an I_2 over schema $\mathbf{S_2}$ such that $\langle I_1, I_2 \rangle \models \Sigma_{12}$ and $\langle I_2, I_3 \rangle \models \Sigma_{23}$.

PROOF OF THE "ONLY IF" DIRECTION. Assume that $\langle I_1, I_3 \rangle \models \Sigma^*_{13}$, that is, $\langle U; I_1, I_3 \rangle \models \Sigma^*_{13}$, for a countably infinite universe U that includes the values in I_1 and I_3. We show that there is an I_2 such that $\langle I_1, I_2 \rangle \models \Sigma_{12}$ and $\langle I_2, I_3 \rangle \models \Sigma_{23}$, for the same choice U of the universe, that is, $\langle U; I_1, I_2 \rangle \models \Sigma_{12}$ and $\langle U; I_2, I_3 \rangle \models \Sigma_{23}$. We take I_2^c such that $\langle I_1, I_2^c \rangle$ is the result of chasing $\langle I_1, \emptyset \rangle$ with Σ_{12}. As we remarked right after Definition 6.5, we have that $\langle I_1, I_2^c \rangle \models \Sigma_{12}$, where the universe is the set of ground terms over $\mathbf{V}$ and $\mathbf{f}$, where $\mathbf{V}$ is the set of values in I_1.

Let $\mathbf{g^0}$ be the collection of concrete functions over U such that $\langle I_1, I_3 \rangle \models T^*_{13}[\mathbf{g} \mapsto \mathbf{g^0}]$. By assumption, the function symbols $\mathbf{f}$ that appear in Σ_{12} are all in $\mathbf{g}$. We denote by $\mathbf{f^0}$ those functions in $\mathbf{g^0}$ that replace function symbols in $\mathbf{f}$.

[4] Although the formula Σ^*_{13} is not an SO tgd under the definition given in this article, it is an SO tgd under the definition given our conference version [Fagin et al. 2004]. This is the difference, mentioned in Footnote 1, between the definition of SO tgds in this article and in the conference version.

We now take I_2 to be the instance that is obtained from I_2^c by instantiating each ground term u of I_2^c with the concrete value that results when we "evaluate" all the function terms in u by using the concrete functions $\mathbf{f^0}$. It is easy to see that I_2 is an instance whose values are in U. Furthermore, it is easy to verify that $\langle I_1, I_2\rangle \models T_{12}[\mathbf{f} \mapsto \mathbf{f^0}]$. In particular, we also have that $\langle U; I_1, I_2\rangle \models \Sigma_{12}$. We will now show that $\langle U; I_2, I_3\rangle \models \Sigma_{23}$.

Let $\mathbf{f}'$ be the collection of all the existentially quantified function symbols in Σ_{23}. We can assume without loss of generality that each f' in $\mathbf{f}'$ appears in a conjunct of Σ_{23}. We show that there are functions $\mathbf{f'^0}$ over U such that $\langle I_2, I_3\rangle \models T_{23}[\mathbf{f}' \mapsto \mathbf{f'^0}]$. In particular, this shows that $\langle U; I_2, I_3\rangle \models \Sigma_{23}$. By construction of Σ_{13}^*, each f' in $\mathbf{f}'$ appears among $\mathbf{g}$ as some g. We take f'^0 to be g^0.

We now show that $\langle I_2, I_3\rangle \models T_{23}[\mathbf{f}' \mapsto \mathbf{f'^0}]$. We need to show that for every conjunct $\forall\mathbf{y}(\psi \rightarrow \gamma)$ in Σ_{23}, we have that $\langle I_2, I_3\rangle \models (\forall\mathbf{y}(\psi \rightarrow \gamma))[\mathbf{f}' \mapsto \mathbf{f'^0}]$. Assume that $\mathbf{a}$ is a sequence of values of I_2 such that $I_2 \models \psi[\mathbf{f}' \mapsto \mathbf{f'^0}, \mathbf{y} \mapsto \mathbf{a}]$. Moreover, we can assume without loss of generality that ψ is of the form

$$R_1(\mathbf{y_1}) \wedge \cdots \wedge R_k(\mathbf{y_k}) \wedge \theta,$$

where the first k literals are the relational atoms of ψ and θ is the conjunction of the equalities in ψ. The variables in $\mathbf{y_p}$, for each p with $1 \leq p \leq k$, appear among the variables in $\mathbf{y}$. (Also recall that the only terms that appear in such relational atoms are variables, by the definition of a second-order tgd.)

Since $I_2 \models \psi[\mathbf{f}' \mapsto \mathbf{f'^0}, \mathbf{y} \mapsto \mathbf{a}]$, it must be the case that I_2 contains a tuple $\mathbf{a_p}$ in R_p, for each p with $1 \leq p \leq k$. Here, $\mathbf{a_p}$ is a subtuple of $\mathbf{a}$, consisting of the sequence of values of $\mathbf{a}$ that replace the variables in $\mathbf{y_p}$. We also have that all the equalities in θ are true when $\mathbf{y}$ is replaced by $\mathbf{a}$ and $\mathbf{f}'$ is replaced by $\mathbf{f'^0}$. That is, for each equality $t = t'$ in θ, we have that $t[\mathbf{f}' \mapsto \mathbf{f'^0}, \mathbf{y} \mapsto \mathbf{a}]$ and $t'[\mathbf{f}' \mapsto \mathbf{f'^0}, \mathbf{y} \mapsto \mathbf{a}]$ represent the same value.

We know that $\langle I_1, I_2^c\rangle$ is the result of chasing $\langle I_1, \emptyset\rangle$ with Σ_{12}, and I_2 is the result of the subsequent "evaluation" of the ground terms in I_2^c by using $\mathbf{f^0}$. Thus, it must be the case that for each tuple $\mathbf{a_p}$, with $1 \leq p \leq k$, there is some conjunct $\forall\mathbf{x_p}(\phi_p(\mathbf{x_p}) \rightarrow R_p(\mathbf{t_p}))$ in Σ_{12} such that there is a homomorphism h_p from this conjunct into I_1 such that $\mathbf{a_p}$ is the result of "evaluating" $h_p(\mathbf{t_p})$ by using $\mathbf{f^0}$. By the definition of a homomorphism from a conjunct into an instance, it must be the case that $I_1 \models \phi_p(h_p(\mathbf{x_p}))$, for every p with $1 \leq p \leq k$. Let h be the union of $h_1, \ldots, h_k$. Thus, h acts on the union $\mathbf{x}$ of the variables in $\mathbf{x_1}, \ldots, \mathbf{x_k}$, and $h(x)$ is defined to be $h_p(x)$, whenever x is among $\mathbf{x_p}$. (The variables in $\mathbf{x_i}$ and $\mathbf{x_j}$ are assumed to be disjoint, for every i and j with $1 \leq i < j \leq k$.) We obtain that

$$I_1 \models (\phi_1(\mathbf{x_1}) \wedge (\mathbf{y_1} = \mathbf{t_1}) \wedge \cdots \wedge \phi_k(\mathbf{x_k}) \wedge (\mathbf{y_k} = \mathbf{t_k}))\,[\mathbf{x} \mapsto h(\mathbf{x}), \mathbf{y} \mapsto \mathbf{a}, \mathbf{f} \mapsto \mathbf{f^0}].$$

Putting this together with the earlier observation that all the equalities in θ are true when $\mathbf{y}$ is replaced by $\mathbf{a}$ and $\mathbf{f}'$ is replaced by $\mathbf{f'^0}$, we obtain that

$$(*)\ \ I_1 \models (\phi_1(\mathbf{x_1}) \wedge (\mathbf{y_1} = \mathbf{t_1}) \wedge \cdots \wedge \phi_k(\mathbf{x_k}) \wedge (\mathbf{y_k} = \mathbf{t_k}) \wedge \theta)$$
$$[\mathbf{x} \mapsto h(\mathbf{x}), \mathbf{y} \mapsto \mathbf{a}, \mathbf{f} \mapsto \mathbf{f^0}, \mathbf{f}' \mapsto \mathbf{f'^0}].$$

Since by assumption the function symbols $\mathbf{g}$ consist precisely of the function symbols in $\mathbf{f}$ along with those in $\mathbf{f}'$, condition (*) is equivalent to:

$$(**)\ I_1 \models (\phi_1(\mathbf{x_1}) \wedge (\mathbf{y_1} = \mathbf{t_1}) \wedge \cdots \wedge \phi_k(\mathbf{x_k}) \wedge (\mathbf{y_k} = \mathbf{t_k}) \wedge \theta)$$
$$[\mathbf{x} \mapsto h(\mathbf{x}), \mathbf{y} \mapsto \mathbf{a}, \mathbf{g} \mapsto \mathbf{g^0}].$$

By the composition algorithm (Step 2), and by the fact that there is some conjunct $\forall \mathbf{x_p}(\phi_p(\mathbf{x_p}) \rightarrow R_p(\mathbf{t_p}))$ in Σ_{12} for each p with $1 \leq p \leq k$, we are guaranteed that Σ_{13}^* contains a conjunct that is obtained from the conjunct $\forall \mathbf{y}(\psi \rightarrow \gamma)$ in Σ_{23}, by replacing each of the literals $R_p(\mathbf{y_p})$ in ψ with the conjunction $\phi_p(\mathbf{x_p}) \wedge (\mathbf{y_p} = \mathbf{t_p})$. Thus, Σ_{13}^* contains the following conjunct:

$$\forall \mathbf{x} \forall \mathbf{y}((\phi_1(\mathbf{x_1}) \wedge (\mathbf{y_1} = \mathbf{t_1}) \wedge \cdots \wedge \phi_k(\mathbf{x_k}) \wedge (\mathbf{y_k} = \mathbf{t_k}) \wedge \theta) \rightarrow \gamma).$$

We know by assumption that $\langle I_1, I_3 \rangle \models T_{13}^*[\mathbf{g} \mapsto \mathbf{g^0}]$. Together with condition (**), this last fact implies that $I_3 \models \gamma[\mathbf{y} \mapsto \mathbf{a}, \mathbf{f}' \mapsto \mathbf{f'^0}]$. We used here that the variables in $\mathbf{x}$ do not appear in γ; we also used the fact that the function symbols that are in $\mathbf{g}$ but not in $\mathbf{f}'$ do not appear in γ (thus, their instantiations do not matter).

Thus, given that $I_2 \models \psi[\mathbf{f}' \mapsto \mathbf{f'^0}, \mathbf{y} \mapsto \mathbf{a}]$ for some arbitrary tuple $\mathbf{a}$ of values, we have concluded that $I_3 \models \gamma[\mathbf{f}' \mapsto \mathbf{f'^0}, \mathbf{y} \mapsto \mathbf{a}]$. It follows that $\langle I_2, I_3 \rangle \models (\forall \mathbf{y}(\psi \rightarrow \gamma))[\mathbf{f}' \mapsto \mathbf{f'^0}]$. This concludes the "only if" direction in the proof.

Proof of the "if" Direction. Assume that $\langle I_1, I_2 \rangle \models \Sigma_{12}$ and $\langle I_2, I_3 \rangle \models \Sigma_{23}$. By Theorem 5.6, if we take a large enough U that includes the active domains of I_1, I_2 and I_3, we have that $\langle U; I_1, I_2 \rangle \models \Sigma_{12}$ and $\langle U; I_2, I_3 \rangle \models \Sigma_{23}$. We will show that $\langle U; I_1, I_3 \rangle \models \Sigma_{13}^*$.

As in the proof of the "only if" direction, let $\mathbf{f}$, $\mathbf{f}'$ and $\mathbf{g}$ denote the collections of existentially quantified function symbols in Σ_{12}, Σ_{23} and Σ_{13}^*, respectively. We know that there are concrete functions $\mathbf{f}$ and $\mathbf{f}'$ over U such that $\langle I_1, I_2 \rangle \models T_{12}[\mathbf{f} \mapsto \mathbf{f^0}]$ and $\langle I_2, I_3 \rangle \models T_{23}[\mathbf{f}' \mapsto \mathbf{f'^0}]$. By the composition algorithm, each function symbol g in $\mathbf{g}$ is either (1) a function symbol f that occurs in Σ_{12} and therefore is in $\mathbf{f}$, or (2) a function symbol f' that occurs in Σ_{23} and therefore is in $\mathbf{f}'$. In the first case, we take g^0 to be f^0. In the second case, we take g^0 to be f'^0. We now show that $\langle I_1, I_3 \rangle \models T_{13}[\mathbf{g} \mapsto \mathbf{g^0}]$.

By the composition algorithm, every conjunct in Σ_{13}^* has the form C, which we define to be

$$\forall \mathbf{x} \forall \mathbf{y}((\phi_1(\mathbf{x_1}) \wedge (\mathbf{y_1} = \mathbf{t_1}) \wedge \cdots \wedge \phi_k(\mathbf{x_k}) \wedge (\mathbf{y_k} = \mathbf{t_k}) \wedge \theta) \rightarrow \gamma),$$

obtained from a conjunct

$$\forall \mathbf{y}((R_1(\mathbf{y_1}) \wedge \cdots \wedge R_k(\mathbf{y_k}) \wedge \theta) \rightarrow \gamma)$$

in Σ_{23} and k conjuncts

$$\forall \mathbf{x_1}(\phi_1(\mathbf{x_1}) \rightarrow R_1(\mathbf{t_1})), \ldots, \forall \mathbf{x_k}(\phi_k(\mathbf{x_k}) \rightarrow R_k(\mathbf{t_k}))$$

in Σ_{12}. Here, the variables $\mathbf{x}$ consist precisely of the variables in $\mathbf{x_1}, \ldots, \mathbf{x_k}$, while the variables $\mathbf{y}$ consist precisely of the variables in $\mathbf{y_1}, \ldots, \mathbf{y_k}$.

We need to show that $\langle I_1, I_3\rangle \models C[\mathbf{g} \mapsto \mathbf{g^0}]$. Assume that $\mathbf{a}$ and $\mathbf{b}$ are sequences of values such that

(i) $I_1 \;\models\; (\phi_1(\mathbf{x_1})\wedge(\mathbf{y_1} = \mathbf{t_1})\wedge\cdots\wedge\phi_k(\mathbf{x_k})\wedge(\mathbf{y_k} = \mathbf{t_k})\wedge\theta)\,[\mathbf{x}\mapsto \mathbf{a}, \mathbf{y}\mapsto \mathbf{b}, \mathbf{g}\mapsto \mathbf{g^0}]$.

To complete the proof, it suffices to show that $I_3 \models \gamma[\mathbf{x} \mapsto \mathbf{a}, \mathbf{y} \mapsto \mathbf{b}, \mathbf{g} \mapsto \mathbf{g^0}]$. This is the same as saying that

$$\text{(ii)} \qquad I_3 \models \gamma[\mathbf{y} \mapsto \mathbf{b}, \mathbf{f}' \mapsto \mathbf{f'^0}].$$

Here we made use of the fact that the variables in $\mathbf{x}$ do not occur in γ. We also made use of the fact that every function symbol g in $\mathbf{g}$ that occurs in γ is some f' in $\mathbf{f}'$, and we had earlier picked g^0 to be equal to f'^0.

We know that $\langle I_1, I_2\rangle \models T_{12}[\mathbf{f} \mapsto \mathbf{f^0}]$. This implies, in particular, that $\langle I_1, I_2\rangle \models \forall\mathbf{x_p}(\phi_p(\mathbf{x_p}) \rightarrow R_p(\mathbf{t_p}))[\mathbf{f} \mapsto \mathbf{f^0}]$, for each p with $1 \leq p \leq k$. From (i) we derive that $I_1 \models \phi_p(\mathbf{x_p})[\mathbf{x} \mapsto \mathbf{a}]$, for each p with $1 \leq p \leq k$. It follows that $I_2 \models R_p(\mathbf{t_p})[\mathbf{x} \mapsto \mathbf{a}, \mathbf{f} \mapsto \mathbf{f^0}]$, for each p with $1 \leq p \leq k$. At the same time, we can also derive from (i) that $\mathbf{y_p}[\mathbf{y} \mapsto \mathbf{b}]$ and $\mathbf{t_p}[\mathbf{x} \mapsto \mathbf{a}, \mathbf{f} \mapsto \mathbf{f^0}]$ represent the same tuple of values, for each p with $1 \leq p \leq k$. Here we made use of the fact that every function symbol g in $\mathbf{g}$ that occurs in $\mathbf{t_p}$ is some f in $\mathbf{f}$, and we had earlier picked g^0 to be equal to f^0. We obtain that

$$I_2 \models (R_1(\mathbf{y_1}) \wedge \cdots \wedge R_k(\mathbf{y_k}))[\mathbf{y} \mapsto \mathbf{b}].$$

Furthermore, from (i), we know that for every equality $t = t'$ in θ, we have that $t[\mathbf{y} \mapsto \mathbf{b}, \mathbf{g} \mapsto \mathbf{g^0}]$ and $t'[\mathbf{y} \mapsto \mathbf{b}, \mathbf{g} \mapsto \mathbf{g^0}]$ represent the same value. (Since θ is from Σ_{23}, the variables in $\mathbf{x}$ do not occur in θ and therefore their values do not matter.) We now make use of the fact that every function symbol g in $\mathbf{g}$ that occurs in θ must occur in $\mathbf{f}'$ as some f', and that we had earlier picked g^0 to be f'^0. We can therefore infer that all the equalities in θ are satisfied when $\mathbf{y}$ is replaced by $\mathbf{b}$ and $\mathbf{f}'$ is replaced by $\mathbf{f'^0}$. We thus obtain the following:

$$\text{(iii)} \qquad I_2 \models (R_1(\mathbf{y_1}) \wedge \cdots \wedge R_k(\mathbf{y_k}) \wedge \theta)\,[\mathbf{y} \mapsto \mathbf{b}, \mathbf{f}' \mapsto \mathbf{f'^0}].$$

We know that $\langle I_2, I_3\rangle \models T_{23}[\mathbf{f}' \mapsto \mathbf{f'^0}]$. This implies, in particular, that $\langle I_2, I_3\rangle \models \forall\mathbf{y}(R_1(\mathbf{y_1}) \wedge \cdots \wedge R_k(\mathbf{y_k}) \wedge \theta \rightarrow \gamma)[\mathbf{f}' \mapsto \mathbf{f'^0}]$. Together with (iii), this implies the earlier statement (ii). This concludes the "if" direction in the proof. □

7.2 The Size of the Composition Formula

Assume that $\mathcal{M}_{13} = \mathcal{M}_{12} \circ \mathcal{M}_{23}$ where $\mathcal{M}_{13} = (\mathbf{S_1}, \mathbf{S_3}, \Sigma_{13})$. We may refer to Σ_{13} as *the composition formula*. When we defined our composition algorithm, we noted that the size of the composition formula that we constructed might be exponential. We now prove an exponential lower bound, which shows that this exponentiality is unavoidable.

PROPOSITION 7.3. *There are schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where Σ_{12} and Σ_{23} are finite sets of full source-to-target tgds, such that if $\mathcal{M}_{12} \circ \mathcal{M}_{23} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$, then every set Σ'_{13} of source-to-target tgds that is logically equivalent to Σ_{13}, and every SO tgd Σ'_{13} that is logically equivalent to Σ_{13}, is of size exponential in the size of $\Sigma_{12} \cup \Sigma_{23}$.*

PROOF. Let $\mathbf{S}_1$ consist of the unary relation symbols $R_1, \ldots, R_n$ and $R'_1, \ldots R'_n$. Let $\mathbf{S}_2$ consist of the unary relation symbols $S_1, \ldots, S_n$, and let $\mathbf{S}_3$ consist of the unary relation symbol T. Let Σ_{12} consist of the full source-to-target tgds $R_i(x) \rightarrow S_i(x)$, for $1 \leq i \leq n$, and the full source-to-target tgds $R'_i(x) \rightarrow S_i(x)$, for $1 \leq i \leq n$. Let Σ_{23} consist of the single full source-to-target tgd $S_1(x) \wedge \cdots \wedge S_n(x) \rightarrow T(x)$. Let Σ_{13} consist of all of the source-to-target tgds of the form $U_1(x) \wedge \cdots \wedge U_n(x) \rightarrow T(x)$, where U_i is either R_i or R'_i, for $1 \leq i \leq n$. It is straightforward to verify that $\mathcal{M}_{12} \circ \mathcal{M}_{23} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$. Let Σ'_{13} be a set of source-to-target tgds that is logically equivalent to Σ_{13}. We shall show that Σ'_{13} contains at least 2^n members. A similar proof show that if Σ'_{13} is an SO tgd, then it contains at least 2^n conjuncts. This is sufficient to prove the theorem.

We first show that the left-hand side of every member of Σ'_{13} must contain at least one of R_i or R'_i, for each i. Assume not; we shall derive a contradiction. Let τ be a member of Σ'_{13} such that there is i for which τ does not contain R_i and τ does not contain R'_i. Let I be a source instance (an $\mathbf{S}_1$ instance) that consists of the facts $R_j(0)$ and $R'_j(0)$ for each $j \neq i$, but where the R_i and R'_i relations are empty. Let J be a target instance (an $\mathbf{S}_3$ instance) where the T relation is empty. It is clear that $\langle I, J \rangle$ satisfies Σ_{13}, since every member of Σ_{13} contains either R_i or R'_i in its left-hand side. However, $\langle I, J \rangle$ does not satisfy τ, since when every variable appearing in the left-hand side of τ takes on the value 0, the left-hand side of τ is satisfied but the right-hand side of τ is not (since the T relation of J is empty). Since $\langle I, J \rangle$ satisfies Σ_{13} but does not satisfy Σ'_{13}, this contradicts the assumption that Σ'_{13} is logically equivalent to Σ_{13}. So indeed, the left-hand side of every member of Σ'_{13} must contain at least one of R_i or R'_i, for each i.

We now show that for each of the 2^n vectors $\mathbf{x} = (x_1, \ldots, x_n)$ where each x_i is either 0 or 1, there is a member $\sigma_{\mathbf{x}}$ of Σ'_{13} such that for each i, the left-hand side of $\sigma_{\mathbf{x}}$ contains R_i precisely if $x_i = 0$, and the left-hand side contains R'_i precisely if $x_i = 1$. Assume not; we shall derive a contradiction. Let $\mathbf{y} = (y_1, \ldots, y_n)$ be a specific 0, 1 vector where this condition is violated, that is, where Σ'_{13} has no such member $\sigma_{\mathbf{y}}$. Let I be a source instance that contains exactly n facts, namely, for each i, the fact $R_i(0)$ when $y_i = 0$ or the fact $R'_i(0)$ when $y_i = 1$, Let J be a target instance where the T relation is empty. We now show that $\langle I, J \rangle$ satisfies every member of Σ'_{13}. Let τ be an arbitrary member of Σ'_{13}. From what we showed earlier, we know that the left-hand side of τ must contain at least one of R_i or R'_i, for each i. Since also τ is not of the form $\sigma_{\mathbf{y}}$, it follows that either there is i such that $y_i = 0$ and τ contains R'_i, or $y_i = 1$ and τ contains R_i. Therefore $\langle I, J \rangle$ satisfies τ, since the left-hand side of τ is never satisfied in I, no matter what the choice is of the variables in the left-hand side of τ. Since τ is an arbitrary member of Σ'_{13}, it follows that $\langle I, J \rangle$ satisfies Σ'_{13}. Now Σ_{13} has a member γ of the form $\sigma_{\mathbf{y}}$. It is easy to see that $\langle I, J \rangle$ does not satisfy γ, and so $\langle I, J \rangle$ does not satisfy Σ_{13}. Since $\langle I, J \rangle$ satisfies Σ'_{13} but does not satisfy Σ_{13}, this contradicts the assumption that Σ'_{13} is logically equivalent to Σ_{13}.

Since Σ'_{13} contains a member $\sigma_{\mathbf{x}}$ for each of the 2^n vectors $\mathbf{x} = (x_1, \ldots, x_n)$ where each x_i is either 0 or 1, and since it is clear that each such member $\sigma_{\mathbf{x}}$ is distinct, it follows that Σ'_{13} contains at least 2^n members. This was to be shown. □

7.3 Failure of the Active Domain Semantics

In our definition of the semantics of SO tgds, we took the universe (which serves as the domain and range of the existentially quantified functions) to be a countably infinite set that includes the active domain. (We later showed that a finite but large enough universe that includes the active domain also suffices.) In this section, we show that if we were to instead take the universe to be simply the active domain, then an SO tgd that results after applying the composition algorithm might have a meaning that is different from that of composition. We also include a discussion on domain independence of SO tgds. Let us refer to our usual semantics as the "infinite universe semantics", and let us refer to the semantics where the universe is taken to be the active domain as the "active domain semantics".

Example 7.4. We consider a slight variation of Example 5.2, with the following schemas $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$. Schema $\mathbf{S}_1$ consists of a single unary relation symbol `Emp` of employees. Schema $\mathbf{S}_2$ consists of a single binary relation symbol `Mgr`, associating each employee with a manager. Schema $\mathbf{S}_3$ consists of a single unary relation symbol `SelfMgr`, intended to store employees who are their own manager. Consider the schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where

$$\Sigma_{12} = \{\forall e\,(\mathtt{Emp}(e) \to \exists m \mathtt{Mgr}(e, m))\} \quad \Sigma_{23} = \{\forall e(\mathtt{Mgr}(e, e) \to \mathtt{SelfMgr}(e))\}.$$

It is easy to verify that the composition algorithm tells us that the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$, where Σ_{13} is the following second-order tgd:

$$\exists f(\forall e(\mathtt{Emp}(e) \wedge (e = f(e)) \to \mathtt{SelfMgr}(e))). \tag{4}$$

In the infinite universe semantics, this formula (4) is equivalent to "Truth", that is, formula (4) is a tautology that holds for every choice of $\langle I_1, I_3\rangle$. This is because we can simply select an arbitrary function f such that $f(e) \neq e$ for every e in the domain. Then the left-hand side $\mathtt{Emp}(e) \wedge (e = f(e))$ is false for every e, and so $\forall e(\mathtt{Emp}(e) \wedge (e = f(e)) \to \mathtt{SelfMgr}(e))$ holds for this f. And indeed, "Truth" is the right answer for the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, as we now show. Let I_1 be an arbitrary instance of schema $\mathbf{S}_1$ and let I_3 be an arbitrary instance of schema $\mathbf{S}_3$. To show that the composition is indeed "Truth", we must show that $\langle I_1, I_3\rangle$ is in the composition. Define the instance I_2 of schema $\mathbf{S}_2$ by letting Bob be some element different from every member of $\mathtt{Emp}^{I_1}$, and letting $\mathtt{Mgr}^{I_2}$ contain all tuples (e, Bob), where e is in $\mathtt{Emp}^{I_1}$. Then $\langle I_1, I_2\rangle \models \Sigma_{12}$ and $\langle I_2, I_3\rangle \models \Sigma_{23}$, and so $\langle I_1, I_3\rangle$ is in the composition, as desired.

We now show that in the active domain semantics, the formula (4) is not equivalent to "Truth". Therefore, the formula (4) given by the composition algorithm, does not have the right meaning, of composition, under the active domain semantics. We need only show that there is an instance I_1 of schema $\mathbf{S}_1$ and an instance I_3 of schema $\mathbf{S}_3$ such that $\langle I_1, I_3\rangle$ does not satisfy (4) in the active domain semantics.

Define I_1 by letting $\mathtt{Emp}^{I_1}$ contain the single element Alice. Define I_3 by letting $\mathtt{SelfMgr}^{I_3}$ be empty. The active domain is {Alice}, and there is only one function with domain and range {Alice}, namely the function f^0 where $f^0(\text{Alice}) = \text{Alice}$. Using this function f^0 (the only function available) for f, the left-hand side $\texttt{Emp}(e) \wedge (e = f(e))$ is satisfied when e is Alice, but the right-hand side $\texttt{SelfMgr}(e)$ is not. So $\langle I_1, I_3\rangle$ does not satisfy the formula (4) in the active domain semantics, which was to be shown.

A formula is said to be *domain independent* if its truth does not depend on the choice of universe, as long as the universe contains the active domain. Fagin [1982] was the first to observe that the safety condition for first-order dependencies (that every universally quantified variable must appear in the left-hand side) makes them domain independent. This comment applies to the (first-order) source-to-target tgds we consider. However, it follows from Example 7.4 that SO tgds are not domain independent. Thus, in this example, let $U' = \{\text{Alice}\}$, which is the active domain, and let $U = \{\text{Alice}, \text{Bob}\}$. It follows easily from the discussion in Example 7.4 that $\langle U; I, J\rangle \models \Sigma_{13}$ but $\langle U'; I, J\rangle \not\models \Sigma_{13}$. Therefore, the SO tgd Σ_{13} is not domain independent.[5] On the other hand, our earlier Theorem 5.6 implies that SO tgds obey a limited form of domain independence: the choice of universe does not matter, as long as it contains the active domain and is sufficiently large.

8. SO TGDS ARE EXACTLY THE NEEDED CLASS

We have introduced SO tgds since (1) every finite set of (first-order) source-to-target tgds is logically equivalent to an SO tgd and (2) SO tgds are closed under composition. We therefore obtain the following theorem.

THEOREM 8.1. *The composition of a finite number of schema mappings, each defined by a finite set of source-to-target tgds, is defined by an SO tgd.*

In this section, we prove a converse to Theorem 8.1. Specifically, we prove the following theorem.

THEOREM 8.2. *Every SO tgd defines the composition of a finite number of schema mappings, each defined by a finite set of source-to-target tgds.*

We note that in Theorem 8.2, the "intermediate" schemas depend on the SO tgd. We now show that Theorem 8.2 gives us the next theorem. This next theorem shows the naturalness and "inevitability" of the class of SO tgds.

THEOREM 8.3. *SO tgds form the smallest class (up to logical equivalence) that contains every source-to-target tgd and is closed under conjunction and composition.*

PROOF. We already noted that every source-to-target tgd is logically equivalent to an SO tgd, and that the conjunction of a pair of SO tgds is logically equivalent to an SO tgd. Also, Theorem 7.2 tells us that the composition of two schema mappings, each defined by an SO tgd, is defined by an SO tgd. These

[5] Sergey Melnik pointed out to us that SO tgds are not necessarily domain independent, using essentially this example.

facts tell us that, up to logical equivalence, the class of SO tgds contains every source-to-target tgd and is closed under conjunction and composition.

We now show that the class of SO tgds is (up to logical equivalence) the smallest such class. Let Y be a class that contains every source-to-target tgd and is closed under conjunction and composition. We must show that for each SO tgd σ there is a member of Y that is logically equivalent to σ. By Theorem 8.2, there are finite sets $\Sigma_{12}, \ldots, \Sigma_{n(n+1)}$ of source-to-target tgds such that σ defines the composition of the mappings given by $\Sigma_{12}, \ldots, \Sigma_{n(n+1)}$. For each i, with $1 \leq i \leq n$, since $\Sigma_{i(i+1)}$ is a finite set of source-to-target tgds, and since Y contains each member of $\Sigma_{i(i+1)}$ and is closed under conjunction, it follows that Y contains the conjunction of the members of $\Sigma_{i(i+1)}$. Since Y is closed under composition, Y contains the composition formula of the schema mappings defined by $\Sigma_{12}, \ldots, \Sigma_{n(n+1)}$. But this composition formula is logically equivalent to σ. So there is a member of Y that is logically equivalent to σ. This was to be shown. □

It remains to prove Theorem 8.2. We shall prove Theorem 8.2 by proving a slightly stronger theorem. Before we state this stronger theorem, we need a definition. The *depth* of a term is as defined in Section 5.2. We define the *nesting depth* of an SO tgd σ to be the largest depth of the terms that appear in σ. For example, let σ be the SO tgd

$$\exists f \exists g (S(x, y) \rightarrow T(x, f(y), g(x, f(y)))).$$

Then σ has nesting depth 2, since the term with the largest depth that appears in σ is $g(x, f(y))$, which has depth 2.

We shall prove the following theorem, which immediately implies Theorem 8.2.

THEOREM 8.4. *Every SO tgd of nesting depth r defines the composition of $r + 1$ schema mappings, each defined by a finite set of source-to-target tgds.*

PROOF. It is instructive to first prove this theorem for some special cases, to get the idea of the construction. Let σ' be the formula $\forall x(S(x) \rightarrow T(f(x), g(x), f(g(x)))$, and let σ be the SO tgd $\exists f \exists g \sigma'$. Thus, σ is

$$\exists f \exists g \forall x (S(x) \rightarrow T(f(x), g(x), f(g(x)))).$$

Define Σ_{12} to consist of the following source-to-target tgds:

$$\forall x (S(x) \rightarrow S_1(x))$$
$$\forall x (S(x) \rightarrow \exists y F_1(x, y))$$
$$\forall x (S(x) \rightarrow \exists y G_1(x, y)).$$

Intuitively, we take S_1 to copy S, we take $F_1(x, y)$ to encode $f(x) = y$, and we take $G_1(x, y)$ to encode $g(x) = y$. Intuitively, the second dependency has the effect of guaranteeing that $f(x)$ is defined whenever $S(x)$ holds, and the third dependency has the effect of guaranteeing that $g(x)$ is defined whenever $S(x)$ holds.

Define Σ_{23} to consist of the following source-to-target tgds:

$$\forall x(S_1(x) \to S_2(x))$$
$$\forall x \forall y(F_1(x, y) \to F_2(x, y))$$
$$\forall x \forall y(G_1(x, y) \to G_2(x, y))$$
$$\forall x \forall y(G_1(x, y) \to \exists z F_2(y, z)).$$

Intuitively, we take S_2 to copy S_1, F_2 to copy F_1, and G_2 to copy G_1. Intuitively, the fourth dependency has the effect of guaranteeing that $f(y)$ is defined for all y in the range of g.

Define Σ_{34} to consist of the following source-to-target tgd:

$$\forall x \forall y \forall y' \forall z((S_2(x) \wedge F_2(x, y) \wedge G_2(x, y') \wedge F_2(y', z)) \to T(y, y', z)). \quad (5)$$

Intuitively, formula (5) says

$$\forall x \forall y \forall y' \forall z((S(x) \wedge (f(x) = y) \wedge (g(x) = y') \wedge (f(y') = z) \to T(y, y', z)). \quad (6)$$

In turn, formula (6) says

$$\forall x(S(x) \to T(f(x), g(x), f(g(x))). \quad (7)$$

Note that formula (7) is exactly the "body" of σ.

We now show that $\langle I_1, I_4 \rangle \models \sigma$ if and only if there are I_2, I_3 such that $\langle I_1, I_2 \rangle \models \Sigma_{12}$, $\langle I_2, I_3 \rangle \models \Sigma_{23}$, and $\langle I_3, I_4 \rangle \models \Sigma_{34}$. This is sufficient to prove the theorem in this special case. Assume first that $\langle I_1, I_4 \rangle \models \sigma$. So there are f^0, g^0 such that $\langle I_1, I_4 \rangle \models \sigma'[f \mapsto f^0, g \mapsto g^0]$. We see from Theorem 5.6 that we can assume without loss of generality that the universe is finite. Define I_2 by taking S_1 to equal S, taking $F_1(a, b)$ to hold in I_2 precisely is $f^0(a) = b$, and taking $G_1(a, b)$ to hold in I_2 precisely if $g^0(a) = b$. Define I_3 by taking S_2 to equal S, taking F_2 to equal F_1, and taking G_2 to equal G_1. Note that I_2 and I_3 are finite, because of our assumption that the universe is finite. It is straightforward to verify that $\langle I_1, I_2 \rangle \models \Sigma_{12}$, $\langle I_2, I_3 \rangle \models \Sigma_{23}$, and $\langle I_3, I_4 \rangle \models \Sigma_{34}$.

Assume now that $\langle I_1, I_2 \rangle \models \Sigma_{12}$, $\langle I_2, I_3 \rangle \models \Sigma_{23}$, and $\langle I_3, I_4 \rangle \models \Sigma_{34}$. Let U (the universe) be a countably infinite set that includes all values that appear in any of I_1, I_2, I_3, or I_4. Define $f^0(a)$ for a in U as follows. If there is some b such that $F_2(a, b)$ holds in I_3, then let $f^0(a)$ be an arbitrary value of b such that $F_2(a, b)$ holds in I_3. (Note that this is reminiscent of our choice of the coloring function in the proof of Theorem 4.6.) For all other a in U, let $f^0(a)$ be an arbitrary member of U. Define $g^0(a)$ for a in U as follows. If there is some b such that $G_2(a, b)$ holds in I_3, then let $g^0(a)$ be an arbitrary value of b such that $G_2(a, b)$ holds in I_3. For all other a in U, let $g^0(a)$ be an arbitrary member of U. It is straightforward to verify that $\langle I_1, I_4 \rangle \models \sigma'[f \mapsto f^0, g \mapsto g^0]$. So $\langle I_1, I_4 \rangle \models \sigma$, as desired.

We note that if we were to apply our composition algorithm to find the result of composing the schema mappings defined by Σ_{12}, Σ_{23} and Σ_{34}, we would obtain a different formula than σ (although this formula is logically equivalent to σ). In particular, when we convert the source-to-target tgds in Σ_{12} and Σ_{23} to SO tgds, we would introduce different Skolem functions for dealing with the tgd $\forall x(S(x) \to \exists y F_1(x, y))$ of Σ_{12} and the tgd $\forall x \forall y(G_1(x, y) \to \exists z F_2(y, z))$ of Σ_{23}. However, it is possible to use the same Skolem function in both cases. The reason is, intuitively, that because of the tgd $\forall x \forall y(F_1(x, y) \rightarrow F_2(x, y))$ of Σ_{23}, the Skolem function needed for the tgd $\forall x \forall y(G_1(x, y) \rightarrow \exists z F_2(y, z))$ of Σ_{23} can simply be an extension to a larger domain of the Skolem function needed for the tgd $\forall x \forall y(G_1(x, y) \rightarrow \exists z F_2(y, z))$ of Σ_{12}.

We now modify our example to allow an equality. Let us take σ_1 to be

$$\exists f \exists g \forall x((S(x) \wedge (f(x) = g(x))) \rightarrow T(f(x), g(x), f(g(x))).$$

Thus, σ_1 is the result of adding the equality $f(x) = g(x)$ to the left-hand side of σ. We then take Σ'_{12} to be Σ_{12}, and Σ'_{23} to be Σ_{23}. We take Σ'_{34} to consist of the following source-to-target tgd:

$$\forall x \forall y \forall z((S_2(x) \wedge F_2(x, y) \wedge G_2(x, y) \wedge F_2(y, z)) \rightarrow T(y, y, z)).$$

Thus, Σ'_{34} is the result of replacing y' by y in Σ_{34}. We then have, similarly to before, that $\langle I_1, I_4 \rangle \models \sigma_1$ if and only if there are I'_2, I'_3 such that $\langle I_1, I'_2 \rangle \models \Sigma'_{12}$, $\langle I'_2, I'_3 \rangle \models \Sigma'_{23}$, and $\langle I'_3, I_4 \rangle \models \Sigma'_{34}$. In fact, we can define I'_2 and I'_3 with the same definitions as we gave for I_2 and I_3 earlier.

We now give the argument in the general case. Let σ be an SO tgd with nesting depth r, with source schema $\mathbf{S}$ and target schema $\mathbf{T}$. Let us write σ as $\exists \mathbf{f} \sigma'$, where σ' is first-order. We must define $r+2$ schemas $\mathbf{S}_1, \ldots, \mathbf{S}_{r+2}$. We let $\mathbf{S}_1$ be $\mathbf{S}$ and let $\mathbf{S}_{r+2}$ be $\mathbf{T}$. For every k-ary relation symbol S of $\mathbf{S}$, and for $2 \leq i \leq r+1$, we let the schema $\mathbf{S}_i$ contain a new k-ary relation symbol S_{i-1}. For every k-ary function symbol f that appears in σ, and for $2 \leq i \leq r+1$, we let the schema $\mathbf{S}_i$ contain a new $(k+1)$-ary relation symbol F_{i-1}. We say that F_{i-1} *represents* f in $\mathbf{S}_i$.

We now define the sets $\Sigma_{i(i+1)}$ for $1 \leq i \leq r+1$. We first define the set Σ_{12}. For every k-ary relation symbol S of $\mathbf{S}$, we let Σ_{12} contain the source-to-target tgd $\forall x_1 \cdots \forall x_k(S(x_1, \ldots, x_k) \rightarrow S_1(x_1, \ldots, x_k))$. Next, we let Σ_{12} contain source-to-target tgds that guarantee, intuitively, that each of the function symbols of σ is defined on the active domain of the instance of schema $\mathbf{S}$. Thus, for every $(k+1)$-ary relation symbol F_1 of $\mathbf{S}_1$ that represents a k-ary function symbol of σ in $\mathbf{S}_2$, and for every combination of choices of atomic formulas from $\mathbf{S}_1$ and every combination of choices of variables $v_1, \ldots, v_k$ that appear in these atomic formulas, we let Σ_{12} contain a source-to-target tgd that guarantees that there is y such that $F_1(v_1, \ldots, v_k, y)$ holds. For example, if F_1 is a ternary relation symbol that represents a binary function symbol of σ in $\mathbf{S}_2$, and if R and S are binary relation symbols of $\mathbf{S}$, then Σ_{12} contains the source-to-target tgd

$$\forall x_1 \forall x_2 \forall x_3 \forall x_4((R(x_1, x_2) \wedge S(x_3, x_4)) \rightarrow \exists y F_1(x_2, x_3, y)).$$

We now define the sets $\Sigma_{i(i+1)}$ for $2 \leq i \leq r$. For every k-ary relation symbol S of $\mathbf{S}$, we let $\Sigma_{i(i+1)}$ contain the source-to-target tgd $\forall x_1 \ldots \forall x_k(S_{i-1}(x_1, \ldots, x_k) \rightarrow S_i(x_1, \ldots, x_k))$. For every k-ary function symbol f that appears in σ, we let $\Sigma_{i(i+1)}$ contain the source-to-target tgd $\forall x_1 \cdots \forall x_k(F_{i-1}(x_1, \ldots, x_{k+1}) \rightarrow F_i(x_1, \ldots, x_{k+1}))$, where F_{i-1} is the relation symbol that represents f in $\mathbf{S}_i$, and F_i is the relation symbol that represents f in $\mathbf{S}_{i+1}$. Next, we let $\Sigma_{i(i+1)}$ contain source-to-target tgds that, intuitively, guarantee, just as we did in the case of Σ_{12}, that each of the function symbols of σ is defined on the active domain of the instance of schema $\mathbf{S}_i$. For example, if G_i is a ternary relation symbol that represents a binary function symbol of

σ in $\mathbf{S}_{i+1}$, and if R is a binary relation symbol of $\mathbf{S}$ (so that R_{i-1} is a binary relation symbol of $\mathbf{S}_i$) and F_{i-1} is a binary relation symbol that represents a unary function symbol of σ in $\mathbf{S}_i$, then $\Sigma_{i(i+1)}$ contains the source-to-target tgd

$$\forall x_1 \forall x_2 \forall x_3 \forall x_4 ((R_{i-1}(x_1, x_2) \wedge F_{i-1}(x_3, x_4)) \rightarrow \exists y \, G_i(x_1, x_4, y)).$$

Note that we did not bother with putting all of these source-to-target tgds into Σ_{23} in our example at the beginning of the proof, since they were not all needed.

Finally, we define the set $\Sigma_{(r+1)(r+2)}$. For each conjunct C_j of σ, where C_j is $\forall \mathbf{x_j}(\phi_j \rightarrow \psi_j)$, we shall define full source-to-target tgds τ'_j (with left-hand side L'_j and right-hand side R'_j) and τ_j (with left-hand side L_j and right-hand side R_j), and we let $\Sigma_{(r+1)(r+2)}$ consist of the tgds τ_j. The difference between τ'_j and τ_j is that in constructing τ'_j, we shall neglect the equalities that appear in C_j; we shall then obtain τ_j by modifying τ'_j to take into account the equalities. We begin by defining, for every term t that appears in C_j, a *terminal variable* v_t and a formula β'_t. If t is a variable x, then v_t and β'_t are both x. If t is the term $f(t_1, \ldots, t_k)$, then we take the terminal variable v_t to be a new variable, and recursively define β'_t to be $F_r(v_{t_1}, \ldots, v_{t_k}, v_t)$, where F_r represents f in $\mathbf{S}_{r+1}$. The left-hand side L'_j of τ'_j is a conjunction of the following formulas:

$S_r(x_1, \ldots, x_p)$, for every atomic formula $S(x_1, \ldots, x_p)$ that appears in ϕ_j
β'_t, for every term t that appears in C_j (including as a subterm).

We can assume without loss of generality that ψ_j consists of a single atomic formula $T(t_1, \ldots, t_m)$. The right-hand side R'_j of τ'_j then is taken to be $T(v_{t_1}, \ldots, v_{t_m})$.

We now describe how we obtain τ_j from τ'_j. Let X_j be the set of equalities $t = t'$ that appear in C_j, and let X'_j be the set of equalities in the transitive, symmetric, reflexive closure of X_j. Then, the terms that are the left-hand side or right-hand side of equalities in C_j form equivalence classes based on X'_j (so that t and t' are in the same equivalence class when the equality $t = t'$ appears in C_j). For each equivalence class, select one term from that equivalence class to be the "representative" of that equivalence class. If some member of the equivalence class is a variable, then let a variable be the representative. If t and t' are in the same equivalence class and if t is the representative of that equivalence class, then form τ_j by replacing every occurrence of $v_{t'}$ in τ'_j by v_t (do this in parallel for each equivalence class). For each term t, denote the formula in τ_j that was obtained from β'_t under this replacement by β_t.

We now show that $\langle I_1, I_{r+2} \rangle \models \sigma$ if and only if there are $I_2, I_3, \ldots, I_{r+1}$ such that $\langle I_i, I_{i+1} \rangle \models \Sigma_{i(i+1)}$ for $1 \leq i \leq r+1$. This is sufficient to prove the theorem. Assume first that $\langle I_1, I_{r+2} \rangle \models \sigma$; we shall show that there are $I_2, I_3, \ldots, I_{r+1}$ such that $\langle I_i, I_{i+1} \rangle \models \Sigma_{i(i+1)}$ for $1 \leq i \leq r+1$. Find $\mathbf{f^0}$ such that

$$\langle I_1, I_{r+2} \rangle \models \sigma'[\mathbf{f} \mapsto \mathbf{f^0}].$$

In particular, for each conjunct C_j of σ', we have

$$\langle I_1, I_{r+2} \rangle \models C_j[\mathbf{f} \mapsto \mathbf{f^0}]. \tag{8}$$

We see from Theorem 5.6 that we can assume without loss of generality that the universe is finite. Define I_2 by taking the S_1 relation of I_2 to equal the S

relation of I_1, for each relation symbol S of $\mathbf{S}$, and taking $F_1(a_1, \ldots, a_k, b)$ to hold in I_2 precisely if $f^0(a_1, \ldots, a_k) = b$, for each function symbol f that appears in σ, where F_1 is the relation symbol that represents f in $\mathbf{S_2}$. Note that F_1 is finite, by our assumption on the universe. For $3 \leq i \leq r+1$, define I_i by taking the S_{i-1} relation of I_i to equal the S relation of I_1, for each S in $\mathbf{S}$, and taking the F_{i-1} relation of I_i to equal the F_1 relation of I_2, for each F_{i-1} that represents a function symbol f of σ in $\mathbf{S_i}$. It is easy to see that by construction of $I_2, \ldots, I_{r+1}$, we have $\langle I_i, I_{i+1} \rangle \models \Sigma_{i(i+1)}$ for $1 \leq i \leq r+1$. We now show that $\langle I_{r+1}, I_{r+2} \rangle \models \Sigma_{(r+1)(r+2)}$.

Let $\widehat{\sigma}$ be the result of replacing each relation symbol S that appears in the left-hand side of a conjunct of σ' by S_r, let $\widehat{C_j}$ be the conjunct of $\widehat{\sigma}$ that corresponds to C_j, and let $\widehat{\phi_j}$ be the left-hand side of $\widehat{C_j}$. Since the S_r relation of I_{r+1} equals the S relation of I_1, it follows from (8) that

$$\langle I_{r+1}, I_{r+2} \rangle \models \widehat{C_j}[\mathbf{f} \mapsto \mathbf{f^0}]. \tag{9}$$

By construction, $F_r(a_1, \ldots, a_k, b)$ holds in I_{r+1} precisely if $f^0(a_1, \ldots, a_k) = b$, for each function symbol f that appears in σ. Let τ_j be the member of $\Sigma_{(r+1)(r+2)}$ that corresponds to the clause C_j of σ. We must show that $\langle I_{r+1}, I_{r+2} \rangle \models \tau_j$.

Let C_j be $\forall \mathbf{x_j}(\phi_j \rightarrow T(t_1, \ldots, t_m))$. Let $\mathbf{v} \mapsto \mathbf{v^0}$ be an assignment of entries of I_{r+1} to the terminal variables where v_t and $v_{t'}$ are assigned the same values if t and t' are in the same equivalence class. Let $\mathbf{x_j} \mapsto \mathbf{x_j^0}$ be the assignment of entries of I_{r+1} to members of $\mathbf{x_j}$ determined by $\mathbf{v} \mapsto \mathbf{v^0}$ (recall that $\mathbf{x_j} \subseteq \mathbf{v}$ since v_x is simply x for variables x). To prove that $\langle I_{r+1}, I_{r+2} \rangle \models \tau_j$, we need only show that if

$$I_{r+1} \models L_j[\mathbf{v} \mapsto \mathbf{v^0}], \tag{10}$$

then

$$I_{r+2} \models R_j[\mathbf{v} \mapsto \mathbf{v^0}]. \tag{11}$$

It is sufficient to restrict to assignments $\mathbf{v} \mapsto \mathbf{v^0}$ of the type we have described, since if t and t' are in the same equivalence class, then at most one of v_t or $v_{t'}$ appears in L_j.

Now L_j is obtained from L'_j by replacing certain variables $v_{t'}$ by variables v_t where t and t' are in the same equivalence class. Since in this case, $v_t^0 = v_{t'}^0$, it follows that (10) is equivalent to the statement

$$I_{r+1} \models L'_j[\mathbf{v} \mapsto \mathbf{v^0}]. \tag{12}$$

By the same argument, it follows that (11) is equivalent to the statement $I_{r+2} \models R'_j[\mathbf{v} \mapsto \mathbf{v^0}]$, that is,

$$I_{r+2} \models T(v_{t_1}, \ldots, v_{t_m})[\mathbf{v} \mapsto \mathbf{v^0}]. \tag{13}$$

We are trying to show that (10) implies (11). Since (10) is equivalent to (12) and since (11) is equivalent to (13), we need only show that (12) implies (13). Assume that (12) holds; we must show that (13) holds.

We now show by induction on depth that if t is a term that appears in C_j, then

$$v_t^0 = t\big[\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}\big]. \tag{14}$$

The base case of depth 0 is immediate, since then t is one of the variables in $\mathbf{x_j}$. We now prove the inductive step. Assume that t is the term $f(t_1, \ldots, t_k)$. Let a_i denote $t_i[\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}]$, for $1 \le i \le k$. Then $t[\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}] = f^0(a_1, \ldots, a_k)$. We must show that $v_t^0 = f^0(a_1, \ldots, a_k)$. Since $F_r(v_{t_1}, \ldots, v_{t_k}, v_t)$ is the conjunct β_t' of L_j', we see from (12) that

$$I_{r+1} \models F_r(v_{t_1}, \ldots, v_{t_k}, v_t)[\mathbf{v} \mapsto \mathbf{v^0}]. \tag{15}$$

By inductive assumption, for $1 \le i \le k$, we have $v_{t_i}^0 = a_i$. By our construction of the F_r relation of I_{r+1}, it then follows from (15) that $v_t^0 = f^0(a_1, \ldots, a_k)$, as desired. This completes the induction, and so completes the proof that (14) holds.

Now $\widehat{\phi_j}$ is the conjunction of certain atomic formulas (all of whose variables are in $\mathbf{x_j}$) and certain equalities between terms. The atomic formulas in $\widehat{\phi_j}$ are guaranteed to hold in I_{r+1} under the assignment $\mathbf{x_j} \mapsto \mathbf{x_j^0}$ because of (12) and the fact that these same atomic formulas appear in L_j'. Whenever $t = t'$ is an equality that appears in $\widehat{\phi_j}$, we have $v_t^0 = v_{t'}^0$ by definition of the assignment $\mathbf{v} \mapsto \mathbf{v^0}$. So by (14), it follows that t and t' take on the same value in the assignment $\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}$. Hence, both the atomic formulas of $\widehat{\phi_j}$ and the equalities of $\widehat{\phi_j}$ are guaranteed to hold in I_{r+1} under the assignment $\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}$. That is,

$$I_{r+1} \models \widehat{\phi_j}\big[\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}\big]. \tag{16}$$

From (9) and (16) and the fact that $\widehat{\phi_j}$ is the left-hand side of $\widehat{C_j}$, it follows that the right-hand side of $\widehat{C_j}$ also holds, that is,

$$I_{r+2} \models T(t_1, \ldots, t_m)\big[\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}\big]. \tag{17}$$

From (14) and (17), we obtain (13), as desired. This completes the proof that $\langle I_i, I_{i+1}\rangle \models \Sigma_{i(i+1)}$ for $1 \le i \le r+1$.

Assume now that $\langle I_i, I_{i+1}\rangle \models \Sigma_{i(i+1)}$ for $1 \le i \le r+1$; we shall show that $\langle I_1, I_{r+2}\rangle \models \sigma$. Let U (the universe) be a countably infinite set that includes all values that appear in one or more of the I_i's. For each k-ary function symbol f that appears in σ, define $f^0(a_1, \ldots, a_k)$ for $a_1, \ldots, a_k$ in U as follows: If there is some b such that $F_r(a_1, \ldots, a_k, b)$ holds in I_{r+1}, then let $f^0(a_1, \ldots, a_k)$ be an arbitrary value of b such that $F_r(a_1, \ldots, a_k, b)$ holds in I_{r+1}. For every other choice of $a_1, \ldots, a_k$ in U, let $f^0(a_1, \ldots, a_k)$ be an arbitrary member of U.

To prove that $\langle I_1, I_{r+2}\rangle \models \sigma$, we need only show that (8) holds for each j. Thus, assume

$$I_1 \models \phi_j\big[\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}\big]; \tag{18}$$

we must show that (17) holds. Since $\widehat{\phi_j}$ is the result of replacing each relation symbol S of ϕ_j by S_r, and since the S_r relation of I_{r+1} equals the S relation of I_1, it follows from (18) that (16) holds.

For simplicity of notation, let us denote $t[\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}]$ by t^0. We now show, by induction on s with $1 \leq s \leq r$, that $\Sigma_{12}, \ldots, \Sigma_{s(s+1)}$ assure that if t is a term $f(t_1, \ldots, t_k)$ of depth s, then $F_s(t_1^0, \ldots, t_k^0, t^0)$ holds in I_{s+1} (and in particular t^0 appears in I_{s+1}).

Let us consider first the base case $s = 1$. In this case, $t_1^0, \ldots, t_k^0$ are each members of the active domain of I_1, and Σ_{12} guarantees that there is b such that $F_1(t_1^0, \ldots, t_k^0, b)$ holds in I_2. By our construction of f^0, it then follows that $F_1(t_1^0, \ldots, t_k^0, t^0)$ holds in I_2.

The inductive step is similar. Let t be the term $f(t_1, \ldots, t_k)$ of depth $s+1$, where $s < r$. Then $t_1, \ldots, t_k$ each have depth at most s, and so by inductive hypothesis, $\Sigma_{12}, \ldots, \Sigma_{s(s+1)}$ assure that $t_1^0, \ldots, t_k^0$ appear in I_{s+1}. Then $\Sigma_{(s+1)(s+2)}$ guarantees additionally that there is b such that $F_{s+1}(t_1^0, \ldots, t_k^0, b)$ holds in I_{s+2}. By our construction of f^0, it then follows that $F_{s+1}(t_1^0, \ldots, t_k^0, t^0)$ holds in I_{s+2}. This completes the induction. Therefore, $\Sigma_{12}, \ldots, \Sigma_{r(r+1)}$ assure that if t is a term $f(t_1, \ldots, t_k)$ that appears in C_j, then $F_r(t_1^0, \ldots, t_k^0, t^0)$ holds in I_{r+1}. That is,

$$I_{r+1} \models F_r(t_1, \ldots, t_k, t)\big[\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}\big]. \tag{19}$$

For each terminal variable v_t, define the assignment $\mathbf{v} \mapsto \mathbf{v^0}$ via (14). Note that this assignment agrees with the assignment $\mathbf{x_j} \mapsto \mathbf{x_j^0}$ if t is a variable. From (14) and (19), we obtain (15).

We now show that (12) holds. The formula L'_j is the conjunction of certain atomic formulas $S_r(x_1, \ldots, x_p)$ and formulas β'_t. The atomic formulas $S_r(x_1, \ldots, x_p)$ in L'_j are guaranteed to hold in I_{r+1} under the assignment $\mathbf{x_j} \mapsto \mathbf{x_j^0}$ (and hence under the assignment $\mathbf{v} \mapsto \mathbf{v^0}$) because of (16) and the fact that these same atomic formulas appear in $\widehat{\phi_j}$. From (15), we see that the formula β'_t holds under this assignment. So (12) holds, as desired.

Because of (16), we know that whenever $t = t'$ is an equality that appears in $\widehat{\phi_j}$, necessarily t and t' take on the same value in the assignment $\mathbf{x_j} \mapsto \mathbf{x_j^0}, \mathbf{f} \mapsto \mathbf{f^0}$. So by (14), we know that v_t and $v_{t'}$ take on the same value in the assignment $\mathbf{v} \mapsto \mathbf{v^0}$. Therefore, it follows as before that (10) is equivalent to (12), and (11) is equivalent to (13). Hence, since (12) holds, it follows that (10) holds,

Since $\langle I_{r+1}, I_{r+2}\rangle \models \tau_j$, and since (10) tells us that the left-hand side of τ_j holds under the assignment $\mathbf{v} \mapsto \mathbf{v^0}$, it follows that the right-hand side of τ_j also holds, that is, (11) holds. Hence, since (11) is equivalent to (13), we know that (13) holds. Since (13) and (14) hold, it follows that (17) holds. This was to be shown. This completes the proof that $\langle I_1, I_{r+2}\rangle \models \sigma$. □

It is interesting to note that every source-to-target tgd in $\Sigma_{i(i+1)}$ in our proof, for $1 \leq i \leq r$, has a single existential quantifier, and every source-to-target tgd in $\Sigma_{(r+1)(r+2)}$ in our proof is full. This may seem counterintuitive, especially if we start with a source-to-target tgd with multiple existential quantifiers, and convert it to an equivalent SO tgd, and then apply the algorithm in the proof of Theorem 8.4. How can we get away with such sets $\Sigma_{i(i+1)}$? Let us consider one more example.

Example 8.5. Let τ be the source-to-target tgd

$$\forall x(S(x) \rightarrow \exists y \exists z T(x, y, z)).$$

Then, an equivalent SO tgd is

$$\exists f \exists g \forall x (S(x) \rightarrow T(x, f(x), g(x))).$$

When we apply the algorithm in the proof of Theorem 8.4, we obtain the following sets Σ_{12} and Σ_{23} of source-to-target tgds. The set Σ_{12} consists of:

$$\begin{gathered}\forall x (S(x) \rightarrow S_1(x))\\ \forall x (S(x) \rightarrow \exists y F_1(x, y))\\ \forall x (S(x) \rightarrow \exists y G_1(x, y)).\end{gathered}$$

The set Σ_{23} consists of:

$$\forall x \forall y \forall z ((S_1(x) \wedge F_1(x, y) \wedge G_1(x, z)) \rightarrow T(x, y, z)).$$

Then the schema mapping defined by the source-to-target tgd τ with two existential quantifiers is equivalent to the composition of the schema mapping defined by Σ_{12} (where each source-to-target tgd has only one existential quantifier) and Σ_{23} (where the only source-to-target tgd is full).

9. CERTAIN-ANSWER ADEQUACY

In this section, we compare and contrast our notion of composition with a different notion of composition that was introduced by Madhavan and Halevy [2003], and further explore their notion.

9.1 Certain-Answer Equivalence

Before introducing Madhavan and Halevy's notion of composition, it is worthwhile to introduce a more general notion, that of *certain-answer equivalence* of schema mappings. This notion is independent of composition, and is a more "relaxed" notion of equivalence for schema mappings than logical equivalence. We will then formulate Madhavan and Halevy's notion of composition in terms of certain-answer equivalence.

Definition 9.1. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st})$ and $\mathcal{M}' = (\mathbf{S}, \mathbf{T}, \Sigma'_{st})$ be schema mappings from $\mathbf{S}$ to $\mathbf{T}$, and let q be a query. We say that $\mathcal{M}$ and $\mathcal{M}'$ are *certain-answer equivalent with respect to q* (and that Σ_{st} and Σ'_{st} are *certain-answer equivalent with respect to q*) if $\mathit{certain}_{\mathcal{M}}(q, I) = \mathit{certain}_{\mathcal{M}'}(q, I)$ for all instances I over $\mathbf{S}$. Let $\mathcal{Q}$ be a class of queries. We say that $\mathcal{M}$ and $\mathcal{M}'$ are *certain-answer equivalent with respect to $\mathcal{Q}$* (and that Σ_{st} and Σ'_{st} are *certain-answer equivalent with respect to $\mathcal{Q}$*) if $\mathcal{M}$ and $\mathcal{M}'$ are certain-answer equivalent with respect to q for each q in $\mathcal{Q}$.

It is clear that if Σ_{st} and Σ'_{st} are logically equivalent, then they are certain-answer equivalent for every class $\mathcal{Q}$ of queries. What about the converse? If $\mathcal{Q}$ is sufficiently rich (e.g., if $\mathcal{Q}$ is the class of conjunctive queries), and if Σ_{st} and Σ'_{st} are certain-answer equivalent with respect to $\mathcal{Q}$, are Σ_{st} and Σ'_{st} necessarily logically equivalent? The next proposition says that the answer is "No.". Thus, certain-answer equivalence is weaker than logical equivalence.

PROPOSITION 9.2. *There are schema mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st})$ and $\mathcal{M}' = (\mathbf{S}, \mathbf{T}, \Sigma'_{st})$, where Σ_{st} and Σ'_{st} are second-order tgds that are not logically equivalent, such that $\mathcal{M}$ and $\mathcal{M}'$ are certain-answer equivalent with respect to conjunctive queries.*

PROOF. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st})$, where $\mathbf{S}$, $\mathbf{T}$, and Σ_{st} are, respectively, $\mathbf{S}_1$, $\mathbf{S}_3$, and the composition formula Σ_{13} from Example 5.2. Thus, Σ_{st} is

$$\begin{aligned} &\exists f (\forall e(\texttt{Emp}(e) \rightarrow \texttt{Mgr}(e, f(e))) \wedge \\ &\quad \forall e(\texttt{Emp}(e) \wedge (e = f(e)) \rightarrow \texttt{SelfMgr}(e))). \end{aligned}$$

Let Σ'_{st} be the second-order tgd that results from dropping the second clause of Σ_{st}. Thus, Σ'_{st} is $\exists f (\forall e(\texttt{Emp}(e) \rightarrow \texttt{Mgr}(e, f(e))))$. Let I be an arbitrary instance of schema $\mathbf{S}$, and let $\langle I, J_0\rangle$ be the result of chasing $\langle I, \emptyset\rangle$ with Σ_{st}. It is easy to see that $\langle I, J_0\rangle$ is also the result of chasing $\langle I, \emptyset\rangle$ with Σ'_{st}. By Theorem 6.8, J_0 is a universal solution for I under both $\mathcal{M}$ and $\mathcal{M}'$. As we noted in Section 2, it was shown in Fagin et al. [2005a] that if q is a conjunctive query (or even a union of conjunctive queries), then $q(J)_\downarrow = \mathit{certain}_{\mathcal{M}}(q, I)$ when J is a universal solution for I. Therefore, $q(J_0)_\downarrow = \mathit{certain}_{\mathcal{M}}(q, I)$, and similarly, $q(J_0)_\downarrow = \mathit{certain}_{\mathcal{M}'}(q, I)$, when q is a conjunctive query. So $\mathit{certain}_{\mathcal{M}}(q, I) = \mathit{certain}_{\mathcal{M}'}(q, I)$. Therefore, Σ_{st} and Σ'_{st} are certain-answer equivalent with respect to conjunctive queries. So we need only show that Σ_{st} and Σ'_{st} are not logically equivalent. Let I_1 and I_3 be as in the proof of Theorem 5.4. As noted there, $\langle I_1, I_3\rangle \not\models \Sigma_{st}$. However, it is easy to see that $\langle I_1, I_3\rangle \models \Sigma'_{st}$. So indeed, Σ_{st} and Σ'_{st} are not logically equivalent. □

By way of contrast, the next proposition says that this difference between certain-answer equivalence and logical equivalence does not arise when we consider sets of source-to-target tgds instead of SO tgds.

PROPOSITION 9.3. *Assume that $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st})$ and $\mathcal{M}' = (\mathbf{S}, \mathbf{T}, \Sigma'_{st})$ are schema mappings where Σ_{st} and Σ'_{st} are sets of source-to-target tgds. Then $\mathcal{M}$ and $\mathcal{M}'$ are certain-answer equivalent with respect to conjunctive queries if and only if Σ_{st} and Σ'_{st} are logically equivalent.*

PROOF. One direction is immediate: logical equivalence implies certain-answer equivalence (with respect to every class of queries, in fact). For the converse, assume that $\mathit{certain}_{\mathcal{M}_{st}}(q, I) = \mathit{certain}_{\mathcal{M}'_{st}}(q, I)$, for every instance I over $\mathbf{S}$ and for every conjunctive query q over $\mathbf{T}$. Let I be an arbitrary instance over $\mathbf{S}$ and let J and J' be universal solutions for I with respect to $\mathcal{M}_{st}$ and $\mathcal{M}'_{st}$, respectively (such universal solutions can be obtained from I by chasing with $\mathcal{M}_{st}$ and $\mathcal{M}'_{st}$, respectively). As noted in the proof of Proposition 9.2, it was shown in Fagin et al. [2005a] that if q is a conjunctive query, then $q(J)_\downarrow = \mathit{certain}_{\mathcal{M}_{st}}(q, I)$. Similarly, $q(J')_\downarrow = \mathit{certain}_{\mathcal{M}'_{st}}(q, I)$. Since by assumption, $\mathit{certain}_{\mathcal{M}_{st}}(q, I) = \mathit{certain}_{\mathcal{M}'_{st}}(q, I)$, it follows that $q(J)_\downarrow = q(J')_\downarrow$. So $q(J)_\downarrow = q(J')_\downarrow$, for every conjunctive query q.

From the last equality we will derive next that J and J' are homomorphically equivalent. That is, we shall show that there is a homomorphism h from J to J' such that $h(c) = c$ for every value c of J that is among the values of I, and there is a similar homomorphism in the other direction. To prove the existence of the first homomorphism, we construct the following canonical conjunctive query q_J associated with J. Let $c_1, \ldots, c_n$ be the distinct elements of J that appear in I and let $d_1, \ldots, d_m$ be all the distinct remaining elements of J (the nulls of J). Let ψ be the conjunction of all atomic formulas over $x_1, \ldots, x_n$, $y_1, \ldots, y_m$ that hold in J when x_i plays the role of c_i and y_j plays the role of d_j, for each i and j. For example, $R(c_2, d_4)$ holds in J if and only if one conjunct in ψ is $R(x_2, y_4)$. Then we define $q_J(x_1, \ldots, x_n)$ to be $\exists y_1 \cdots \exists y_m \psi$.

It is easy to see that the tuple $(c_1, \ldots, c_n)$ is in $q_J(J)_\downarrow$. Since $q_J(J)_\downarrow = q_J(J')_\downarrow$, it follows that $(c_1, \ldots, c_n)$ is in $q_J(J')_\downarrow$. Hence, there must be a valuation from the variables $x_1, \ldots, x_n$, $y_1, \ldots, y_m$ of q_J to values of J' such that all atoms of ψ are mapped homomorphically (i.e., preserving relations) into tuples of J', and moreover x_i is mapped to c_i for each i. Given the construction of q_J from J, we obtain a homomorphism h from J to J' such that $h(c_i) = c_i$ for each i. Since $c_1, \ldots, c_n$ are all the values of J that occur in I, we obtain that h is a homomorphism from J to J' such that $h(c) = c$ for every value c of J that occurs in I. A symmetric argument shows the existence of a similar homomorphism from J' to J.

We now show that Σ_{st} and Σ'_{st} are logically equivalent. Let I and K be arbitrary instances over $\mathbf{S}$ and $\mathbf{T}$. Assume that $\langle I, K\rangle \models \Sigma_{st}$. In other words, K is a solution for I with respect to $\mathcal{M}_{st}$. Let J and J' be universal solutions for I, with respect to $\mathcal{M}_{st}$ and $\mathcal{M}'_{st}$, respectively. The universality of J implies that there is a homomorphism g from J to K such that $g(c) = c$ for every value c of J that occurs in I. Moreover, we have shown that J and J' are homomorphically equivalent. In particular, there is a homomorphism h' from J' to J such that $h'(c') = c'$ for every value c' of J' that occurs in I. Composing homomorphisms yields homomorphisms. We thus obtain a homomorphism k from J' to K that moreover satisfies $k(c') = c'$ for every value c' of J' that is in I. Furthermore, we have that $\langle I, J'\rangle \models \Sigma'_{st}$, since J' is in particular a solution for I with respect to $\mathcal{M}'_{st}$.

Finally, we use the following property of source-to-target tgds, which can be easily verified: if $\langle I, J'\rangle$ satisfies a source-to-target tgd τ and there is a homomorphism from J' to K that maps values of I into themselves, then $\langle I, K\rangle$ also satisfies τ. Applying this property to the above I, J' and K and the set Σ'_{st} of source-to-target tgds, we obtain that $\langle I, K\rangle \models \Sigma'_{st}$. We have shown that if $\langle I, K\rangle \models \Sigma_{st}$, then $\langle I, K\rangle \models \Sigma'_{st}$. We thus proved that Σ_{st} logically implies Σ'_{st}. A symmetric argument shows the reverse implication. □

We now define Madhavan and Halevy's notion of composition using our terminology and notation. Let $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ be schema mappings, with $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$. Assume that $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ and $\mathcal{M}'_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma'_{13})$ are schema mappings, where $\mathcal{M}_{13}$ is the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$. Let q be a query. We say that Σ'_{13} is *certain-answer adequate for q (with respect to $\mathcal{M}_{12}$, $\mathcal{M}_{23}$)* if Σ_{13} and Σ'_{13} are certain-answer equivalent with respect to q. Let $\mathcal{Q}$ be a class of queries. We say that Σ'_{13} is *certain-answer adequate for $\mathcal{Q}$ (with respect to $\mathcal{M}_{12}$, $\mathcal{M}_{23}$)* if Σ'_{13} is certain-answer adequate for q (with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$) for each q in $\mathcal{Q}$. Thus, Σ'_{13} is certain-answer adequate for $\mathcal{Q}$ (with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$) precisely if $\mathit{certain}_{\mathcal{M}_{13}}(q, I) = \mathit{certain}_{\mathcal{M}'_{13}}(q, I)$

for all instances I over $\mathbf{S}_1$ and all queries q in $\mathcal{Q}$. Intuitively, certain-answer adequacy says that the certain answers of queries in $\mathcal{Q}$ (over $\mathbf{S}_3$) with respect to an instance I of $\mathbf{S}_1$ are the same whether we use the schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ or the schema mapping $\mathcal{M}'_{13}$ to arrive at the answers. Madhavan and Halevy used certain-answer adequacy as their notion of composition. They were especially interested in the case where $\mathcal{Q}$ is the class of conjunctive queries.

The next proposition follows immediately from the definition of certain-answer adequacy.

PROPOSITION 9.4. *Let $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ be schema mappings, and let Σ_{13} be the composition formula. Let q be an arbitrary query. Then Σ_{13} is certain-answer adequate for q with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$.*

Note that in Proposition 9.4, we make no assumption on $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, such as that Σ_{12} and Σ_{13} are sets of source-to-target tgds.

We now show that in some situations, there exists Σ'_{13} that is certain-answer adequate but not logically equivalent to the composition formula Σ_{13}. This is why we use the word "adequate": logically inequivalent choices may both be adequate for the job.

THEOREM 9.5. *There are schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where Σ_{12} and Σ_{23} are finite sets of source-to-target tgds, and there are two logically inequivalent formulas that are each certain-answer adequate for conjunctive queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$.*

PROOF. Let $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ be as in Example 5.2. Let Σ_{13} and Σ'_{13} be, respectively, Σ_{st} and Σ'_{st} from the proof of Proposition 9.2. Let $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ and $\mathcal{M}'_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma'_{13})$. The proof of Proposition 9.2 shows that Σ_{13} is the composition formula, that $\mathcal{M}_{13}$ and $\mathcal{M}'_{13}$ are certain-answer equivalent with respect to conjunctive queries, and Σ_{13} and Σ'_{13} are logically inequivalent. So Σ_{13} and Σ'_{13} are logically inequivalent formulas that are each certain-answer adequate for conjunctive queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$. This proves the theorem. □

9.2 Dependence of Certain-Answer Adequacy on the Class of Queries

In this section, we explore the dependence of certain-answer adequacy on the class of queries. We prove the following results:

(A) A formula may be certain-answer adequate for conjunctive queries but not for conjunctive queries with inequalities.

(B) A formula may be certain-answer adequate for conjunctive queries with inequalities but not for all first-order queries.

(C) A formula is certain-answer adequate for all first-order queries if and only if it is (logically equivalent to) the composition formula. It follows that if a formula is certain-answer adequate for all first-order queries, then it is certain-answer adequate for all queries.

Since the composition formula is certain-answer adequate for all queries, we see from (B) that there is a scenario where there are two different formulas (namely, the formula guaranteed by (B) and the composition formula) that are both certain-answer adequate for conjunctive queries with inequalities. This strengthens the result we already had (Theorem 9.5) that there is a scenario where there are two different formulas that are both certain-answer adequate for conjunctive queries.

We now prove these results. We begin by proving (A).

Theorem 9.6. *There are schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where Σ_{12} and Σ_{23} are finite sets of source-to-target tgds, and where there is a formula that is certain-answer adequate for conjunctive queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$ but not certain-answer adequate for conjunctive queries with inequalities with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$.*

Proof. Let $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ be as in Example 5.2. Let σ be

$$\begin{aligned}&\exists f(\forall e(\texttt{Emp}(e) \rightarrow \texttt{Mgr}(e, f(e))) \wedge \\ &\quad \forall e(\texttt{Emp}(e) \wedge (e = f(e)) \rightarrow \texttt{SelfMgr}(e))).\end{aligned}$$

As shown in Example 5.2, σ is the composition formula. In the proof of Proposition 9.2, we gave a formula (in fact, an SO tgd) that was denoted by Σ_{st} that is certain-answer equivalent to σ with respect to conjunctive queries but is not logically equivalent to σ. We now give another formula (in this case, not an SO tgd) such that σ and σ' are certain-answer equivalent with respect to conjunctive queries, but are not certain-answer equivalent with respect to conjunctive queries with inequalities. This is sufficient to prove the theorem, since it implies that σ' is certain-answer adequate for conjunctive queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$ but not certain-answer adequate for conjunctive queries with inequalities with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$.

Let σ' be the following formula:

$$\begin{aligned}&\exists f(\forall e(\texttt{Emp}(e) \rightarrow \texttt{Mgr}(e, f(e))) \wedge \\ &\quad \forall e(\texttt{Emp}(e) \wedge (e = f(e)) \rightarrow \texttt{SelfMgr}(e)) \wedge \\ &\quad \forall e \forall e'(\texttt{Emp}(e) \wedge \texttt{Emp}(e') \wedge (f(e) = f(e')) \rightarrow (e = e'))).\end{aligned}$$

Thus, the only difference between σ and σ' is that σ' requires that the existentialized function f be one-to-one on the domain of the Emp relation.

We now show that σ and σ' are certain-answer equivalent with respect to conjunctive queries. Let $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \sigma)$ and $\mathcal{M}'_{13} = (\mathbf{S}_1, \mathbf{S}_3, \sigma')$. Let I be an instance over $\mathbf{S}_1$, and let q be a conjunctive query. We must show that $\mathit{certain}_{\mathcal{M}_{13}}(q, I) = \mathit{certain}_{\mathcal{M}'_{13}}(q, I)$. Since σ' logically implies σ, it follows easily that $\mathit{certain}_{\mathcal{M}_{13}}(q, I) \subseteq \mathit{certain}_{\mathcal{M}'_{13}}(q, I)$. We now show the reverse inclusion. Let t be a tuple in $\mathit{certain}_{\mathcal{M}'_{13}}(q, I)$. That is,

$$t \in \bigcap\{q(J) : \langle I, J\rangle \in \mathrm{Inst}(\mathcal{M}'_{13})\} \tag{20}$$

It is easy to see that t contains no nulls. Let $\langle I, J_0\rangle$ be a result of chasing $\langle I, \emptyset\rangle$ with σ. Since the chase process associates a unique null with each syntactically different term generated during the chase process, it follows that J_0 satisfies not just σ but also σ'. So from (20), it follows that $t \in q(J_0)$. Since t contains no nulls, we have $t \in q(J_0)_{\downarrow}$. Since J_0 is a universal solution for I under $\mathcal{M}_{13}$ (by

Proposition 6.8) and q is a conjunctive query, it follows as before that $q(J_0)_\downarrow = \mathit{certain}_{\mathcal{M}_{13}}(q, I)$. Therefore, $t \in \mathit{certain}_{\mathcal{M}_{13}}(q, I)$, as desired.

We show next that σ and σ' are not certain-answer equivalent with respect to conjunctive queries with inequalities. Let q be the query $\exists y_1 \exists y_2((y_1 \neq y_2) \wedge \mathtt{Mgr}(x_1, y_1) \wedge \mathtt{Mgr}(x_2, y_2)$. Then q is a conjunctive query with inequalities. Let I be $\{\mathtt{Emp}(\text{Alice}), \mathtt{Emp}(\text{Bob})\}$. Since σ' forces $f(\text{Alice}) \neq f(\text{Bob})$, it follows easily that

$$\mathit{certain}_{\mathcal{M}'_{13}}(q, I) = \{(\text{Alice}, \text{Bob}), (\text{Bob}, \text{Alice})\}.$$

However, $\mathit{certain}_{\mathcal{M}_{13}}(q, I) = \emptyset$, since one solution J (for which $\langle I, J\rangle \models \sigma$) is

$$\{\mathtt{Mgr}(\text{Alice}, \text{Alice}), \mathtt{Mgr}(\text{Bob}, \text{Alice}), \mathtt{SelfMgr}(\text{Alice})\},$$

where there is no tuple that satisfies q. Therefore,

$$\mathit{certain}_{\mathcal{M}_{13}}(q, I) \neq \mathit{certain}_{\mathcal{M}'_{13}}(q, I). \tag{21}$$

Hence, σ and σ' are not certain-answer equivalent with respect to q, which is a conjunctive query with inequalities. This was to be shown □

Theorem 9.6 says that a formula may be certain-answer adequate for conjunctive queries but not certain-answer adequate for conjunctive queries with inequalities. This brings up the natural question as to whether a formula that is certain-answer adequate for conjunctive queries with inequalities is necessarily certain-answer adequate for all queries, or at least for all first-order queries. The next theorem says that this is not the case.

THEOREM 9.7. *There are schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where Σ_{12} and Σ_{23} are finite sets of source-to-target tgds, and where there is a formula that is certain-answer adequate for conjunctive queries with inequalities with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$ but not certain-answer adequate for all first-order queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$.*

PROOF. Let $\mathbf{S}_1$ be a schema with three unary relation symbols $\mathtt{A}_1, \mathtt{B}_1, \mathtt{C}_1$; let $\mathbf{S}_2$ be a schema with three unary relation symbols $\mathtt{A}_2, \mathtt{B}_2, \mathtt{C}_2$; and let $\mathbf{S}_3$ be a schema with three unary relation symbols $\mathtt{A}_3, \mathtt{B}_3, \mathtt{C}_3$. Consider now the schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where

$$\begin{aligned} \Sigma_{12} = \{ & \forall x(\mathtt{A}_1(x) \to \mathtt{A}_2(x)), \\ & \forall x(\mathtt{B}_1(x) \to \mathtt{B}_2(x)), \\ & \forall x(\mathtt{C}_1(x) \to \mathtt{C}_2(x)) \} \end{aligned} \qquad \begin{aligned} \Sigma_{23} = \{ & \forall x(\mathtt{A}_2(x) \to \mathtt{A}_3(x)), \\ & \forall x(\mathtt{B}_2(x) \to \mathtt{B}_3(x)), \\ & \forall x(\mathtt{C}_2(x) \to \mathtt{C}_3(x)) \} \end{aligned}$$

Then the composition formula Σ_{13} is

$$\forall x((\mathtt{A}_1(x) \to \mathtt{A}_3(x)) \wedge (\mathtt{B}_1(x) \to \mathtt{B}_3(x)) \wedge (\mathtt{C}_1(x) \to \mathtt{C}_3(x)).$$

Let Σ'_{13} be the conjunction of Σ_{13} with $\forall x(\mathtt{C}_3(x) \to \exists y((\mathtt{A}_3(y) \vee \mathtt{B}_3(y))$. We shall show that Σ'_{13} is certain-answer adequate for conjunctive queries with inequalities with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$ but not certain-answer adequate for all first-order queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$.

We first show that Σ'_{13} is certain-answer adequate for conjunctive queries with inequalities with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$. Let q be a conjunctive query with inequalities. Define $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ and $\mathcal{M}'_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma'_{13})$. We must show that

$$\mathit{certain}_{\mathcal{M}_{13}}(q, I) = \mathit{certain}_{\mathcal{M}'_{13}}(q, I). \tag{22}$$

If either the $\mathtt{A}_1$ or $\mathtt{B}_1$ relation of I is nonempty, then it is easy to see that for every J, we have $\langle I, J\rangle \models \Sigma_{13}$ if and only if $\langle I, J\rangle \models \Sigma'_{13}$, which implies that (22) holds. So we are done unless the $\mathtt{A}_1$ and $\mathtt{B}_1$ relations of I are empty. Assume that the $\mathtt{A}_1$ and $\mathtt{B}_1$ relations of I are empty. If q contains some conjunct of the form $\mathtt{A}_3(x)$ or some conjunct of the form $\mathtt{B}_3(x)$, then it is easy to see that the left-hand side and right-hand side of (22) are both empty, and hence equal. Therefore, assume that q contains only inequalities and formulas of the form $\mathtt{C}_3(x)$. By the safety condition on conjunctive queries with inequalities, for every inequality $x \neq y$ that appears in q, the formulas $\mathtt{C}_3(x)$ and $\mathtt{C}_3(y)$ appear in q. Let q^* be the result of replacing each occurrence of $\mathtt{C}_3$ by $\mathtt{C}_1$. It is easy to see that both the left-hand side and right-hand side of (22) contain precisely the tuples t such that $q^*(t)$ holds in I. So once again, the left-hand side and right-hand side of (22) are equal. This concludes the proof that Σ'_{13} is certain-answer adequate for conjunctive queries with inequalities with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$.

Finally, we show that Σ'_{13} is not certain-answer adequate for all first-order queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$. Let q be the first-order query $\mathtt{C}_3(x) \rightarrow \exists y((\mathtt{A}_3(y) \vee \mathtt{B}_3(y))$. We shall show that Σ'_{13} is not certain-answer adequate for q with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$. Let $I = \{\mathtt{A}_3(0)\}$. Then

$$\mathit{certain}_{\mathcal{M}'_{13}}(q, I) = \{0\}, \tag{23}$$

since for each J where $\langle I, J\rangle \models \Sigma'_{13}$, we have $0 \in q(J)$. However,

$$\mathit{certain}_{\mathcal{M}_{13}}(q, I) = \emptyset, \tag{24}$$

since if we let $J = \{\mathtt{C}_3(0)\}$, then we see that $\langle I, J\rangle \models \Sigma_{13}$ and $q(J) = \emptyset$. It follows from (23) and (24) that

$$\mathit{certain}_{\mathcal{M}_{13}}(q, I) \neq \mathit{certain}_{\mathcal{M}'_{13}}(q, I).$$

Thus, Σ'_{13} is not certain-answer adequate for q with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$. This concludes the proof. □

Theorems 9.6 and 9.7 both demonstrate that there is a formula that is certain-answer adequate for all queries in a class $\mathcal{Q}_1$ but not for a richer class $\mathcal{Q}_2$. The next theorem will be used to prove that once $\mathcal{Q}_1$ consists of all first-order queries, there is no such class $\mathcal{Q}_2$.

THEOREM 9.8. *Let $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ be schema mappings. The only formula (up to logical equivalence) that is certain-answer adequate for all first-order queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$ is the composition formula.*

PROOF. We shall show that there is at most one formula that is certain-answer adequate for all first-order queries. Since, by Proposition 9.4, we know that the composition formula is certain-answer adequate for every query with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$, this is sufficient to prove the theorem.

Assume that there are two formulas Σ'_{13} and Σ''_{13} that are each certain-answer adequate for all first-order queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$. We must show that Σ'_{13} and Σ''_{13} are logically equivalent. Assume that $\langle I, J\rangle \models \Sigma'_{13}$; we shall show that $\langle I, J\rangle \models \Sigma''_{13}$.

Let Σ_{13} be the composition formula. Assume that $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$. Define $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$, $\mathcal{M}'_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma'_{13})$, and $\mathcal{M}''_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma''_{13})$. Let q be an arbitrary first-order query. Since Σ'_{13} is certain-answer adequate for q, we have

$$certain_{\mathcal{M}_{13}}(q, I) = certain_{\mathcal{M}'_{13}}(q, I). \tag{25}$$

Similarly, since Σ''_{13} is certain-answer adequate for q, we have

$$certain_{\mathcal{M}_{13}}(q, I) = certain_{\mathcal{M}''_{13}}(q, I). \tag{26}$$

It follows from (25) and (26) that

$$certain_{\mathcal{M}'_{13}}(q, I) = certain_{\mathcal{M}''_{13}}(q, I). \tag{27}$$

Let $c_1, \ldots, c_n$ be the distinct elements of I that appear in J, and let $d_1, \ldots, d_m$ be the distinct remaining elements of J. Let ψ_1 be the formula that is the conjunction of all atomic formulas and negations of atomic formulas over $x_1, \ldots, x_n, y_1, \ldots, y_m$ that hold in J when x_i plays the role of c_i, and y_j plays the role of d_j, for each i, j. For example, if $R(c_3, d_9)$ holds in J, then one conjunct is $R(x_3, y_9)$. If $R(c_3, d_9)$ does not hold in J, then one conjunct is $\neg R(x_3, y_9)$. Let ψ_2 be the conjunction of all of the inequalities $x_i \neq x_j$ for $i \neq j$, all of the inequalities $y_i \neq y_j$ for $i \neq j$, and all of the inequalities $x_i \neq y_j$. Let ψ_3 be the formula

$$\forall x((x = x_1) \vee (x = x_2) \vee \cdots \vee (x = x_n) \vee (x = y_1) \vee (x = y_2) \vee \cdots \vee (x = y_m)).$$

Let ϕ' be the formula $\psi_1 \wedge \psi_2 \wedge \psi_3$, let ϕ be the formula $\exists y_1 \cdots \exists y_m \phi'$, and let q be the query $\neg\phi$. Then $(c_1, \ldots, c_n)$ is not in $certain_{\mathcal{M}'_{13}}(q, I)$, since $J \models \phi'[x_1 \mapsto c_1, \ldots, x_n \mapsto c_n]$. So by (27), we know that $(c_1, \ldots, c_n)$ is not in $certain_{\mathcal{M}''_{13}}(q, I)$. This means that there is J' where $\langle I, J'\rangle \models \Sigma''_{13}$ such that $J' \models \phi'[x_1 \mapsto c_1, \ldots, x_n \mapsto c_n]$. But by the design of ϕ, we know that J' is isomorphic to J under an isomorphism that maps each member of I onto itself. Hence, $\langle I, J'\rangle$ is isomorphic to $\langle I, J\rangle$. Since $\langle I, J'\rangle \models \Sigma''_{13}$, and since $\langle I, J'\rangle$ is isomorphic to $\langle I, J\rangle$, it follows that $\langle I, J\rangle \models \Sigma''_{13}$. This was to be shown. □

COROLLARY 9.9. *Let $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ be schema mappings, and let ϕ be a formula that is certain-answer adequate for all first-order queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$. Then ϕ is certain-answer adequate for every query with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$.*

PROOF. Let ϕ be a formula that is certain-answer adequate for all first-order queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$. Theorem 9.8 says that ϕ is the composition formula. Proposition 9.4 then says that ϕ is certain-answer adequate for every query with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$. □

Note that in Theorem 9.8 and Corollary 9.9, as in Proposition 9.4, we make no assumption on $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, such as that Σ_{12} and Σ_{13} are sets of source-to-target tgds.

An examination of the proof of Theorem 9.8 shows that the proof actually shows the stronger result that the only formula that is certain-answer adequate for all ∀∃ first-order queries is the composition formula.

9.3 The Inadequacy of Finite Sets of TGDs

Our earlier Proposition 4.4 tells us that in some cases, the composition is not definable by any finite set of source-to-target tgds. A natural question at this point is whether a finite set of tgds is always sufficient for certain-answer adequacy for conjunctive queries when the schema mappings $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ are finite sets of tgds. Our next result answers this question negatively.

THEOREM 9.10. *There are schema mappings $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where Σ_{12} and Σ_{23} are finite sets of source-to-target tgds, where no finite set of source-to-target tgds is certain-answer adequate for conjunctive queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$.*

PROOF. The proof is based on the proof of Proposition 4.4. As in that proof, the schema mappings that we use to prove the theorem are $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ of Example 2.3. Let Σ_{13} be the composition formula, and let $\mathcal{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$. Let I_1 be as in the proof of Proposition 4.4. Let q_m be the conjunctive query

$$\exists y(\texttt{Enrollment}(y, x_1) \wedge \cdots \wedge \texttt{Enrollment}(y, x_m)).$$

It follows from the proof of Proposition 4.4 that $(c_1, \ldots, c_m) \in \mathit{certain}_{\mathcal{M}_{13}}(q_m, I_1)$. Let Σ_{13}^{fin} be a finite set of source-to-target tgds, and let $\mathcal{M}_{13}^{\text{fin}} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13}^{\text{fin}})$. It follows from the proof of Proposition 4.4 that $(c_1, \ldots, c_m) \notin \mathit{certain}_{\mathcal{M}_{13}^{\text{fin}}}(q_m, I_1)$ if m is sufficiently large. So $\mathit{certain}_{\mathcal{M}_{13}}(q_m, I_1) \neq \mathit{certain}_{\mathcal{M}_{13}^{\text{fin}}}(q_m, I_1)$ if m is sufficiently large. Hence, Σ_{13}^{fin} is not certain-answer adequate for conjunctive queries with respect to $\mathcal{M}_{12}, \mathcal{M}_{23}$. □

We note that Madhavan and Halevy gave an example where an infinite set of tgds is certain-answer adequate for conjunctive queries but no finite subset of it is. The above Theorem 9.10 shows a stronger negative example where no finite set of tgds whatsoever suffices for certain-answer adequacy.

9.4 Contrasting Our Approach with Madhavan and Halevy's Approach

We close this section with some comparisons between our notion of composition and Madhavan and Halevy's notion (which we *call certain-answer* adequacy). Our approach has the following advantages over theirs:

(1) Our approach is, we feel, more natural than theirs, in that the intent in both cases is to capture the notion of composition, and we do that directly.
(2) Our approach is sufficiently powerful to capture theirs, in that the composition formula is always certain-answer adequate for every query (Proposition 9.4).
(3) Certain-answer adequacy is defined relative to a class $\mathcal{Q}$ of queries, whereas the composition formula is not. The class $\mathcal{Q}$ of queries matters, as demonstrated by Theorem 9.6, which says that there is a formula that is

Table I. Differences between SO tgds and source-to-target tgds

Schema Mapping Language	Compose?	Model Checking	Universal Solution	Certain Answers for CQs	Equivalence Same as Certain-answer Equivalence for CQs?
source-to-target tgds	No	PTIME	PTIME	PTIME	Yes
SO tgds	Yes	NP, can be NP-complete	PTIME	PTIME	No

certain-answer adequate for conjunctive queries but not certain-answer adequate for conjunctive queries with inequalities.

(4) There may be logically inequivalent formulas that are each certain-answer adequate for conjunctive queries (Theorem 9.5), whereas the composition formula is unique (up to logical equivalence).

(5) An infinite set Σ of tgds may be required for certain-answer adequacy for conjunctive queries (Theorem 9.10), whereas an SO tgd, which is finite, suffices to define the composition. Madhavan and Halevy give a representation for this infinite set Σ that is sometimes finite. We note that SO tgds always serve as such a finite representation.

10. CONCLUSIONS

We have introduced what we believe to be the right notion of the composition of two schema mappings. We have also introduced second-order tgds, which are a generalization of finite sets of source-to-target tgds, but with function symbols and equalities. We believe that second-order tgds are the right language for specifying and composing schema mappings. We show that second-order tgds are robust, in that the composition of mappings, each given by a second-order tgd, is also given by a second-order tgd. By contrast, when the mappings are each given by a finite set of source-to-target tgds, their composition may not be definable by even an infinite set of source-to-target tgds. We show that second-order tgds form the smallest class (up to logical equivalence) that contains every source-to-target tgd and is closed under conjunction and composition. We also show that second-order tgds possess good properties for data exchange. As in the case of data exchange with a finite set of source-to-target tgds, a universal solution for a fixed data exchange setting, specified with a second-order tgd, can be computed in polynomial time. Consequently, the certain answers for conjunctive queries can also be computed in polynomial time. Table I summarizes some of the differences between second-order tgds and source-to-target tgds.

ACKNOWLEDGMENTS

The authors thank Sergey Melnik, Moshe Y. Vardi, and the anonymous referees for helpful suggestions.

REFERENCES

Abiteboul, S. and Duschka, O. M. 1998. Complexity of answering queries using materialized views. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*. ACM, New York, 254–263.

ABITEBOUL, S., HULL, R., AND VIANU, V. 1995. *Foundations of Databases*. Addison-Wesley, Reading, MA.

BEERI, C. AND VARDI, M. Y. 1984a. A proof procedure for data dependencies. *J. ACM. 31*, 4, 718–741.

BEERI, C. AND VARDI, M. Y. 1984b. Formal systems for tuple and equality generating dependencies. *SIAM J. Comput. 13*, 1, 76–98.

BERNSTEIN, P. A. 2003. Applying model management to classical meta-data problems. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*. 209–220.

CHANDRA, A. K. AND HAREL, D. 1982. Structure and complexity of relational queries. *J. Comput. Syst. Sci. 25*, 1, 99–128.

DAWAR, A. 1998. A restricted second order logic for finite structures. *Inf. Comput. 143*, 2, 154–174.

EBBINGHAUS, H.-D. AND FLUM, J. 1999. *Finite Model Theory*, 2nd Edition. Springer-Verlag, New York.

ENDERTON, H. B. 2001. *A Mathematical Introduction to Logic*, 2nd Edition. Academic Press, Orlando, FL.

FAGIN, R. 1974. Generalized first-order spectra and polynomial-time recognizable sets. In *Complexity of Computation, SIAM-AMS Proceedings, Vol. 7*, R. M. Karp, Ed. 43–73.

FAGIN, R. 1982. Horn clauses and database dependencies. *J. ACM 29*, 4 (Oct.), 952–985.

FAGIN, R., KOLAITIS, P. G., MILLER, R. J., AND POPA, L. 2005a. Data exchange: semantics and query answering. *Theoret. Comput. Sci. 336*, 89–124. (Preliminary version in *Proceedings of the 2003 International Conference on Database Theory*, pp. 207–224.)

FAGIN, R., KOLAITIS, P. G., AND POPA, L. 2005b. Data exchange: Getting to the core. *ACM Trans. Datab. Syst. 30*, 1 (March), 174–210. (A preliminary version appeared in Proceedings of the Symposium on Principles of Database Systems (PODS 2003). ACM New York, 90–101).

FAGIN, R., KOLAITIS, P. G., POPA, L., AND TAN, W.-C. 2004. Composing schema mappings: Second-order dependencies to the rescue. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*. ACM, New York, 83–94.

FEDER, T. AND VARDI, M. 1998. The computational structure of monotone monadic SNP and constraint satisfaction: A study through datalog and group theory. *SIAM J. Comput. 28*, 57–104. (Preliminary version in *Proc. 25th ACM Symp. on Theory of Computing*, May 1993, pp. 612–622.)

GAREY, M., JOHNSON, D. S., AND STOCKMEYER, L. J. 1976. Some simplified NP-complete graph problems. *Theoret. Comput. Sci. 1*, 237–267.

HALEVY, A. Y., IVES, Z. G., MORK, P., AND TATARINOV, I. 2003. Piazza: Data management infrastructure for semantic web applications. In *Proceedings of the International World Wide Web Conference*. 556–567.

IMMERMAN, N. 1999. *Descriptive Complexity*. Springer-Verlag, New York.

LENZERINI, M. 2002. Data integration: A theoretical perspective. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*. ACM, New York, 233–246.

MADHAVAN, J. AND HALEVY, A. Y. 2003. Composing mappings among data sources. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. 572–583.

MILLER, R. J., HAAS, L. M., AND HERNÁNDEZ, M. 2000. Schema mapping as query discovery. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. 77–88.

POPA, L., VELEGRAKIS, Y., MILLER, R. J., HERNANDEZ, M. A., AND FAGIN, R. 2002. Translating web data. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. 598–609.

VASSILIADIS, P., SIMITSIS, A., AND SKIADOPOULOS, S. 2002. On the logical modeling of ETL processes. In *Proceedings of the International Conference on Advanced Information Systems Engineering (CAiSE)*. 782–786.

Received November 2004; revised July 2005; accepted September 2005