Local Transformations and Conjunctive-Query Equivalence

Ronald Fagin
IBM Research – Almaden
fagin@us.ibm.com

Phokion G. Kolaitis
UC Santa Cruz & IBM Research – Almaden
kolaitis@cs.ucsc.edu

ABSTRACT

Over the past several decades, the study of conjunctive queries has occupied a central place in the theory and practice of database systems. In recent years, conjunctive queries have played a prominent role in the design and use of schema mappings for data integration and data exchange tasks. In this paper, we investigate several different aspects of conjunctive-query equivalence in the context of schema mappings and data exchange.

In the first part of the paper, we introduce and study a notion of a local transformation between database instances that is based on conjunctive-query equivalence. We show that the chase procedure for GLAV mappings (that is, schema mappings specified by source-to-target tuple-generating dependencies) is a local transformation with respect to conjunctive-query equivalence. This means that the chase procedure preserves bounded conjunctive-query equivalence, that is, if two source instances are indistinguishable using conjunctive queries of a sufficiently large size, then the target instances obtained by chasing these two source instances are also indistinguishable using conjunctive queries of a given size. Moreover, we obtain polynomial bounds on the level of indistinguishability between source instances needed to guarantee indistinguishability between the target instances produced by the chase. The locality of the chase extends to schema mappings specified by a second-order tuple-generating dependency (SO tgd), but does not hold for schema mappings whose specification includes target constraints.

In the second part of the paper, we take a closer look at the composition of two GLAV mappings. In particular, we break GLAV mappings into a small number of well-studied classes (including LAV and GAV), and complete the picture as to when the composition of schema mappings from these various classes can be guaranteed to be a GLAV mapping, and when they can be guaranteed to be conjunctive-query equivalent to a GLAV mapping.

We also show that the following problem is decidable: given a schema mapping specified by an SO tgd and a GLAV mapping, are they conjunctive-query equivalent? In contrast, the following problem is known to be undecidable: given a schema mapping specified by an SO tgd and a GLAV mapping, are they logically equivalent?

PODS '12, May 21–23, 2012, Scottsdale, Arizona, USA.

Categories and Subject Descriptors

H.2.5 [**Database Management**]: Heterogeneous Databases—*data translation*; H.2.4 [**Database Management**]: Systems—*relational databases*

General Terms

Algorithms, Theory

Keywords

local transformations, continuity, conjunctive queries, schema mappings, chase, composition

1. Introduction

Conjunctive queries have played a major role in both the theory and practice of relational database systems since the early days of the relational data model. They are now ubiquitous in the study of data inter-operability tasks, such as data exchange and data integration (see the overviews [Len02, Kol05, ABLM10]). In particular, conjunctive queries play a key role in the design of schema-mapping languages, that is, high-level, declarative languages whose formulas are used to describe the relationship between two database schemas, often referred to as the source schema and the target (or global) schema. For example, GLAV mappings, the most widely used and extensively studied schema mappings, are specified by a finite set of source-to-target tuple-generating dependencies (s-t tgds) each of which, intuitively, asserts that some conjunctive query over the source schema is contained in some conjunctive query over the target schema. Furthermore, much of the study of query answering in data exchange and data integration has focused on the problem of computing the certain answers of conjunctive queries over the target schema in the case of data exchange (or over the global schema in the case of data integration).

In a different yet related direction of research, conjunctive queries have been also used to formulate a notion of equivalence between schema mappings that is a relaxation of the classical notion of logical equivalence. Specifically, schema mappings $\mathbf{M_1}$ and $\mathbf{M_2}$ are said to be *conjunctive-query equivalent* (or, in short, *CQ-equivalent*) if for every conjunctive query q over the target schema and for every source instance I, the certain answers of q on I w.r.t. $\mathbf{M_1}$ coincide with the certain answers of q on I w.r.t. $\mathbf{M_2}$. In [FKNP08], CQ-equivalence was studied in the context of schema-mapping optimization. In particular, CQ-equivalence was shown to coincide with logical equivalence for GLAV mappings, but to be a strict relaxation of logical equivalence for schema mappings involving target constraints, as well as for schema mappings specified by

second-order tuple-generating dependencies (SO tgds). Subsequent investigations of CQ-equivalence in the context of schema-mapping optimization include [PSS11] and [FPSS11]. Prior to all these investigations, however, a notion of composition of two schema mappings based on CQ-equivalence was introduced and studied in [MH03]. More recently, a notion of an inverse of schema mapping based on CQ-equivalence was introduced and studied in [APRR09].

Our goal in this paper is to investigate several different aspects of conjunctive-query equivalence in the context of data exchange, as well as in the context of composing schema mappings. We begin by introducing and studying the notion of a CQ-*local* transformation between database instances, a notion that is based on *bounded conjunctive-query equivalence*. Intuitively, a CQ-local transformation has the property that if two instances are indistinguishable using conjunctive queries of a sufficiently large size, then their images under the transformation are also indistinguishable using conjunctive queries of a given size. Formally, a transformation $\mathcal{F}$ between database instances is CQ-*local* if for every positive integer n, there is a positive integer N such that if I_1 and I_2 are instances that satisfy the same Boolean conjunctive queries with at most N variables, then their images $\mathcal{F}(I_1)$ and $\mathcal{F}(I_2)$ satisfy the same Boolean conjunctive queries with at most n variables.

We show that if $\mathbf{M}$ is a GLAV mapping, then the chase procedure w.r.t. $\mathbf{M}$ is a CQ-local transformation. As a matter of fact, we give two different proofs of this result. The first proof entails combining the main technical result in Rossman's proof of the preservation-under-homomorphisms theorem in the finite [Ros08] with a result from the full, unpublished version of [ABFL04] to the effect that the chase transformation for GLAV mappings is local in a sense of first-order equivalence. This proof yields an N that is a stack of exponentials in n, because this type of blow-up already occurs in Rossman's proof [Ros08], and no smaller bounds are presently known. We therefore give a different and direct proof of the CQ-locality of the chase procedure for a GLAV mapping that also yields an N that is bounded by a polynomial in the size of n. In fact, the degree of the polynomial is equal to the maximum arity of the relation symbols of the target schema. We also point out that the CQ-locality of the chase procedure extends to schema mappings specified by SO tgds, but does not hold for schema mappings whose specification includes target constraints.

In the second part of the paper, we take a closer look at the composition of two GLAV mappings. In [FKPT05], it was shown that the composition of two GLAV mappings is guaranteed to be logically equivalent to a schema mapping specified by an SO tgd, but may not be logically equivalent to any GLAV mapping. In fact, as also shown in [FKPT05], the composition of two GLAV mappings may not even be CQ-equivalent to any GLAV mapping. It is also known, however, that the state of affairs is different for the important cases of GAV and LAV mappings. A GAV (global-as-view) mapping is a schema mapping specified by a finite set of s-t tgds whose right-hand side is a single atom, while a LAV (local-as-view) mapping is a schema mapping specified by a finite set of s-t tgds whose left-hand side is a single atom in which no variable occurs more than once. As regards GAV mappings, it was shown in [FKPT05] that the composition of two GAV mappings is guaranteed to be logically equivalent to a GAV mapping; furthermore, the composition of a GAV mapping with a GLAV mapping is guaranteed to be logically equivalent to a GLAV mapping. As regards LAV mappings, it was shown in [AFM10] that the composition of two LAV mappings is guaranteed to be logically equivalent to a GLAV mapping (in fact, to a LAV mapping). Here, we generalize this result by showing that the composition of a GLAV mapping with a LAV mapping is guaranteed to be logically equivalent to a GLAV mapping. After this, we consider the class of *extended* LAV mappings, which are schema mappings specified by a finite set of s-t tgds whose left-hand side is a single atom in which a variable may occur more than once. Clearly, extended LAV mappings form a proper extension of the class of LAV mappings. We show that the composition of a GLAV mapping with an extended LAV mapping is guaranteed to be CQ-equivalent to a GLAV mapping (such a composition may not be logically equivalent to any GLAV mapping [FKPT05]). With the aid of these two results, we complete the picture as to when the composition of schema mappings taken from the classes of GAV, LAV, extended LAV, and arbitrary GLAV mappings can be guaranteed to be a GLAV mapping, and when it can be guaranteed to be CQ-equivalent to a GLAV mapping.

Finally, we show that the following problem is decidable: given a schema mapping specified by an SO tgd and a GLAV mapping, are they CQ-equivalent? In contrast, as shown in [FPSS11] by building on results from [APR09], the following problem is undecidable: given a schema mapping specified by an SO tgd and a GLAV mapping, are they logically equivalent?

2. Preliminaries

A *schema* $\mathbf{R}$ is a finite sequence $\langle R_1, \ldots, R_k \rangle$ of relation symbols, where each R_i has a fixed arity. An *instance* I over $\mathbf{R}$, or an $\mathbf{R}$-*instance*, is a sequence $(R_1^I, \ldots, R_k^I)$, where each R_i^I is a finite relation of the same arity as R_i. We shall often use R_i to denote both the relation symbol and the relation R_i^I that instantiates it. A *fact* of an instance I (over $\mathbf{R}$) is an expression $R_i^I(v_1, \ldots, v_m)$ (or simply $R_i(v_1, \ldots, v_m)$), where R_i is a relation symbol of $\mathbf{R}$ and $(v_1, \ldots, v_m) \in R_i^I$. The expression $(v_1, \ldots, v_m)$ is also sometimes referred to as a *tuple* of R_i. An instance is often identified with its set of facts. An entry in a tuple of an instance I is an *element* or *value* from I, and the set of elements from I is the *active domain* of I.

Next, we define the concepts of *homomorphism* and *homomorphic equivalence*. Let I_1 and I_2 be instances over a schema $\mathbf{R}$. A function h is a *homomorphism* from I_1 to I_2 if for every relation symbol R in $\mathbf{R}$ and every tuple $(a_1, \ldots, a_n) \in R^{I_1}$, we have that $(h(a_1), \ldots, h(a_n)) \in R^{I_2}$. In data exchange, it is often convenient to assume the presence of two kinds of values, namely *constants* and *(labeled) nulls*, and to assume as part of the definition of a homomorphism h that $h(c) = c$ for every constant c; however, we do not make that assumption in this paper. We use the notation $I_1 \rightarrow I_2$ to denote that there is a homomorphism from I_1 to I_2. Since we do not assume that a homomorphism necessarily maps each constant into itself, it is sometimes important to specify that a homomorphism h *respects* I for some instance I, which means that $h(x) = x$ for every element x of I. If there is a homomorphism from I_1 to I_2 that respects I, then we may write $I_1 \stackrel{I}{\rightarrow} I_2$. We say that I_1 is *homomorphically equivalent* to I_2, written $I_1 \leftrightarrow I_2$, if $I_1 \rightarrow I_2$ and $I_2 \rightarrow I_1$. The *core* of an instance K is the smallest subinstance of K that is homomorphically equivalent to K. If there are multiple cores of K, then they are all isomorphic [HN92].

Schema mappings A *schema mapping* is a triple $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ and $\mathbf{T}$ are schemas with no relation symbols in common, and Σ is a set of constraints (typically, formulas in some logic) that describe the relationship between $\mathbf{S}$ and $\mathbf{T}$. We say that $\mathbf{M}$ is *specified by* Σ. We refer to $\mathbf{S}$ as the *source schema*, and $\mathbf{T}$ as the *target schema*. Similarly, we refer to $\mathbf{S}$-instances as *source instances*, and $\mathbf{T}$-instances as *target instances*. We say that schema mappings $\mathbf{M}_1$ and $\mathbf{M}_2$ are *logically equivalent* if the constraints that specify $\mathbf{M}_1$ are logically equivalent to the constraints that specify $\mathbf{M}_2$.

If I is a source instance and J is a target instance such that the pair (I, J) satisfies Σ (written $(I, J) \models \Sigma$), then we say that J is a *solution* of I w.r.t. $\mathbf{M}$. We say that J is a *universal solution for* I w.r.t. $\mathbf{M}$ [FKMP05] if J is a solution for I and for every solution J' for I, we have $J \xrightarrow{I} J'$.

An *atom* is an expression $R(x_1, ..., x_n)$, where R is a relation symbol and $x_1, \ldots, x_n$ are variables that are not necessarily distinct. A *source-to-target tuple-generating dependency (s-t tgd)* is a first-order sentence of the form $\forall \mathbf{x}(\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi(\mathbf{x}, \mathbf{y}))$, where $\varphi(\mathbf{x})$ is a conjunction of atoms over $\mathbf{S}$, each variable in $\mathbf{x}$ occurs in at least one atom in $\varphi(\mathbf{x})$, and $\psi(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms over $\mathbf{T}$ with variables in $\mathbf{x}$ and $\mathbf{y}$. For simplicity, we will often suppress writing the universal quantifiers $\forall \mathbf{x}$ in the above formula. We refer to $\varphi(\mathbf{x})$ as the *left-hand side*, or *premise*, and $\exists \mathbf{y}\psi(\mathbf{x}, \mathbf{y})$ as the *right-hand side*, or *conclusion*. Another name for s-t tgds is *global-and-local-as-view* (GLAV) constraints (see [Len02]). They contain several important special cases, which we now define.

A GAV *(global-as-view)* constraint is an s-t tgd in which the conclusion is a single atom with no existentially quantified variables, that is, it is of the form $\forall \mathbf{x}(\varphi(\mathbf{x}) \rightarrow P(\mathbf{x}))$, where $P(\mathbf{x})$ is an atom over the target schema.

There are several competing notions of a LAV *(local-as-view)* constraint. The definition we shall use is that a LAV constraint is an s-t tgd of the form $\forall \mathbf{x}(Q(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi(\mathbf{x}, \mathbf{y}))$, where $Q(\mathbf{x})$ is a single atom over the source schema and no repeated variables in $Q(\mathbf{x})$ are allowed. This is the notion of LAV used by Arocena, Fuxman, and Miller [AFM10], for their result that the composition of LAV mappings is logically equivalent to a LAV mapping. Another notion of LAV is obtained by dropping the restriction that there are no repeated variables in the premise $Q(\mathbf{x})$. We shall refer to such constraints as *extended LAV*. In a number of papers, including [ABFL04, FKMP05, Fag07, FKPT11], our notion of "extended LAV" is called simply "LAV", and in [FKPT11], our notion of "LAV" is called "strict LAV". Note also that there is yet another notion of "LAV" in the literature, which is defined even more strictly than our definition, by requiring that all variables in $\mathbf{x}$ appear in the conclusion.

We refer to a schema mapping specified entirely by a finite set of GLAV (respectively, GAV, LAV, extended LAV) constraints as a GLAV (respectively, GAV, LAV, extended LAV) mapping.

On occasion, we will also consider schema mappings whose specification also includes target constraints. A target *equality generating dependency (egd)* is a first-order sentence that is of the form $\forall \mathbf{x}(\varphi(\mathbf{x}) \rightarrow (x_i = x_j))$, where $\varphi(\mathbf{x})$ is a conjunction of atoms over $\mathbf{T}$, each variable in $\mathbf{x}$ occurs in at least one atom in $\varphi(\mathbf{x})$, and x_i, x_j are among the variables in $\mathbf{x}$. A target *tuple-generating dependency (tgd)* is a first-order sentence of the form $\forall \mathbf{x}(\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi(\mathbf{x}, \mathbf{y}))$, where both $\varphi(\mathbf{x})$ and $\psi(\mathbf{x}, \mathbf{y})$ are conjunctions of atoms over $\mathbf{T}$, and each variable in $\mathbf{x}$ occurs in at least one atom in $\varphi(\mathbf{x})$. A target *full tgd* is a target tgd whose conclusion has no existential quantifiers.

We shall also make use of *second-order tgds*, or *SO tgds*. These were introduced in [FKPT05], where it was shown that SO tgds are exactly what is needed to specify the composition of an arbitrary number of GLAV mappings. Before we formally define SO tgds, we need to define *terms*.

Given collections $\mathbf{x}$ of variables and $\mathbf{f}$ 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.

DEFINITION 2.1. 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:

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

where

1. Each member of $\mathbf{f}$ is a function symbol.
2. Each ϕ_i is a conjunction of
   • atoms $S(y_1, ..., y_k)$, 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 atoms $T(t_1, ..., t_l)$, 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.

Each subformula $\forall \mathbf{x_i}(\phi_i \rightarrow \psi_i)$ is a *tgd part* of the SO tgd. □

As an example, in a personnel database, where $\texttt{Emp}(e)$ means that e is an employee, $\texttt{Mgr}(e, e')$ means that e' is the manager of e, and $\texttt{SelfMgr}(e)$ means that e is his own manager, we might have the following SO tgd, where, intuitively $f(e)$ is the manager of e:

$$\exists f(\forall e(\texttt{Emp}(e) \rightarrow \texttt{Mgr}(e, f(e))) \wedge \forall e(\texttt{Emp}(e) \wedge (e = f(e)) \rightarrow \texttt{SelfMgr}(e))). \quad (1)$$

We now give the definition (from [FKPT05]) of the composition of schema mappings. Let $\mathbf{M}_{12}$ = ($\mathbf{S}_1$, $\mathbf{S}_2$, Σ_{12}) and let $\mathbf{M}_{23}$ = ($\mathbf{S}_2$, $\mathbf{S}_3$, Σ_{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 $\mathbf{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ is a *composition of* $\mathbf{M}_{12}$ *and* $\mathbf{M}_{23}$ if for every $\mathbf{S}_1$-instance I_1 and every $\mathbf{S}_3$-instance I_3 we have that $(I_1, I_3) \models \Sigma_{13}$ if and only if there is an $\mathbf{S}_2$-instance I_2 such that $(I_1, I_2) \models \Sigma_{12}$ and $(I_2, I_3) \models \Sigma_{23}$. We may then write $\mathbf{M}_{13} = \mathbf{M}_{12} \circ \mathbf{M}_{23}$, and $\Sigma_{13} = \Sigma_{12} \circ \Sigma_{23}$.

Chase The *chase procedure* [ABU79, MMS79] has been used in a number of settings over the years, and several variants of the chase procedure have been considered. In this paper, we use the variant described in [FKNP08], which is sometimes called the *naive chase* or the *parallel chase*.. The basic idea of the naive chase procedure on a source instance I with a GLAV mapping $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is that for every s-t tgd $\forall \mathbf{x}(\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi(\mathbf{x}, \mathbf{y}))$ in Σ and for every tuple $\mathbf{a}$ of values from the active domain of I, such that $I \models \varphi(\mathbf{a})$, we add all facts in $\psi(\mathbf{a}, \mathbf{N})$ to the output of the chase procedure, where $\mathbf{N}$ is a tuple of new, distinct values (usually called labeled nulls) interpreting the existential quantified variables $\mathbf{y}$. Note that in the naive chase, we add these facts whether or not there is already a tuple $\mathbf{b}$ of values such that $\psi(\mathbf{a}, \mathbf{b})$ is in the current output of the chase procedure. From now on, we refer to the naive chase procedure as simply the *chase procedure* or the *chase*, and we write $chase_{\mathcal{M}}(I)$ or $chase_{\Sigma}(I)$ to denote the result of applying the chase procedure on the instance I. It is shown in [FKMP05] that $chase_{\mathcal{M}}(I)$ is a universal solution of I w.r.t. $\mathbf{M}$.

Note that all of our results hold no matter which variant of the chase procedure is used, because for a fixed GLAV mapping, the results of all variants are homomorphically equivalent,

A *conjunctive query* over a schema $\mathbf{R}$ is a formula of the form $\exists \mathbf{y}\phi(\mathbf{x}, \mathbf{y})$ where $\phi(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms over $\mathbf{R}$. If $\mathbf{x}$ is empty (that is, if every variable is existentially quantified) then we call the conjunctive query *Boolean*.

Let $\mathbf{M}$ be a schema mapping, q a k-ary query, for $k \geq 0$, over the target schema $\mathbf{T}$, and I a source instance. The *certain answers*

of q with respect to I, denoted by $certain_{\mathbf{M}}(q, I)$, is the set of all k-tuples t of elements from I such that for every solution J for I with respect to $\mathbf{M}$, we have that $t \in q(J)$. In symbols,

$$certain_{\mathbf{M}}(q, I) = \bigcap \{q(J) : J \text{ is a solution for } I \text{ w.r.t. } \mathbf{M}\}.$$

If q is a Boolean query, then $certain_{\mathbf{M}}(q, I) = true$ precisely when $q(J) = true$, for every solution J for I w.r.t. $\mathbf{M}$. If $\mathbf{M}$ is specified by Σ, then we may write $certain_{\Sigma}(q, I)$ instead of $certain_{\mathbf{M}}(q, I)$.

We shall make use of the following theorem from [FKMP05].

THEOREM 2.2 ([FKMP05]). *Let $\mathbf{M}$ be an arbitrary schema mapping and I an arbitrary source instance such that I has a universal solution U with respect to $\mathbf{M}$. Let q be a conjunctive query.*[1] *Then $certain_{\mathbf{M}}(q, I) = q(U)_{\downarrow}$, which is the result of evaluating q on U and then keeping only those tuples formed entirely of values from I.*

3. Local Transformations

We begin by introducing a unifying notion of a *local transformation*.

DEFINITION 3.1. Let $D = \{D_n : n \geq 1\}$ be a family of binary relations between instances such that for each n, we have that $D_{n+1} \subseteq D_n$, and for each I_1, I_2, if $D_n(I_1, I_2)$, then I_1 and I_2 are instances over the same schema.

Let $\mathbf{S}$ and $\mathbf{T}$ be schemas. If $\mathcal{F}$ is a function "preservfrom the class of $\mathbf{S}$-instances to the class of $\mathbf{T}$-instances, then we say that $\mathcal{F}$ is a *D-local transformation* if for every positive integer n, there is a positive integer N such that for all $\mathbf{S}$-instances I_1 and I_2 with $D_N(I_1, I_2)$, we have that $D_n(\mathcal{F}(I_1), \mathcal{F}(I_2))$. □

If $\mathcal{F}$ is D-local, then for every positive integer n, there is a positive integer N such that for all $m \geq N$ and for all $\mathbf{S}$-instances I_1 and I_2 with $D_m(I_1, I_2)$, we have that $D_n(\mathcal{F}(I_1), \mathcal{F}(I_2))$. This is so because it follows from Definition 3.1 that $D_m \subseteq D_N$ when $m \geq N$. Before we give our case of greatest interest, we need another definition.

DEFINITION 3.2. Assume that I_1 and I_2 are $\mathbf{S}$-instances over a schema $\mathbf{S}$, and let n be a positive integer. We say that I_1 and I_2 are *CQ_n-equivalent*, and write $I_1 \equiv_n^{cq} I_2$, if I_1 and I_2 satisfy the same Boolean conjunctive queries with at most n variables. □

The binary relations $\equiv_n^{cq}$, $n \geq 1$, give rise to the family

$$\mathrm{CQ} \quad = \quad \{\equiv_n^{cq}:\ n \geq 1\}.$$

Our case of greatest interest for D-locality in Definition 3.1 is when $D = \mathrm{CQ}$. Thus, a transformation $\mathcal{F}$ is CQ-local if for every positive integer n, there is a positive integer N such that for all instances I_1 and I_2, if $I_1 \equiv_N^{cq} I_2$, then $\mathcal{F}(I_1) \equiv_n^{cq} \mathcal{F}(I_2)$.

We shall make use of the following simple lemma, which follows easily from the fact that the $\equiv_n^{cq}$ relationship between two instances depends only on their homomorphism equivalence classes.

LEMMA 3.3. *Assume that $I_1 \leftrightarrow I_1'$, $I_2 \leftrightarrow I_2'$, and $I_1 \equiv_n^{cq} I_2$. Then $I_1' \equiv_n^{cq} I_2'$.*

We now point out that CQ-locality can be viewed as a type of uniform continuity with respect to a natural metric that has been studied in graph theory. We begin with a measure of *similarity* between two instances.

[1] This theorem is shown in [FKMP05] to hold a little more generally: not just for conjunctive queries, but also for unions of conjunctive queries.

DEFINITION 3.4. If I_1 and I_2 are $\mathbf{S}$-instances for some schema $\mathbf{S}$, and n is a positive integer, then

$$\mathrm{sim}(I_1, I_2) = \min\{|C| : ((C \to I_1) \text{ and } (C \not\to I_2)) \text{ or } ((C \not\to I_1) \text{ and } (C \to I_2))\},$$

where $|C|$ is size of the active domain of C. □

We have the following simple proposition.

PROPOSITION 3.5. *Assume that I_1 and I_2 are $\mathbf{S}$-instances over a schema $\mathbf{S}$, and n is a positive integer. Then $I_1 \equiv_n^{cq} I_2$ if and only if* $\mathrm{sim}(I_1, I_2) > n$.

This proposition is an immediate consequence of the Chandra-Merlin Theorem [CM77]. Indeed, for every positive integer n and for all instances I_1 and I_2, the following are equivalent:

- I_1 and I_2 satisfy the same conjunctive queries with at most n variables.
- For every instance C with at most n elements, we have that $C \to I_1$ if and only if $C \to I_2$.

EXAMPLE 3.6. For every positive integer m, let C_m be the undirected cycle with m elements and let K_m be the clique with m elements. It is easy to verify that the following are true:

1. $\mathrm{sim}(C_{2i+1}, C_{2j+1}) = 2i + 1$, for $1 \leq i < j$.
2. $\mathrm{sim}(K_2, C_{2j+1}) = 2j + 1$, for $j \geq 1$
3. $\mathrm{sim}(K_i, K_j) = i + 1$, for $i < j$. In particular, we have that $\mathrm{sim}(K_2, K_j) = 3$, for $j \geq 3$. □

Define a distance measure d between $\mathbf{S}$-instances by letting

$$d(I_1, I_2) = \frac{1}{\mathrm{sim}(I_1, I_2)}.$$

In particular, $d(I_1, I_2) = 0$ if and only if $\mathrm{sim}(I_1, I_2) = \infty$, which holds if and only if I_1 and I_2 are homomorphically equivalent. Moreover, if $I_1 \leftrightarrow I_1'$ and $I_2 \leftrightarrow I_2'$, then $d(I_1, I_2) = d(I_1', I_2')$. Therefore, d can be viewed as a distance between $\leftrightarrow$-equivalence classes of $\mathbf{S}$-instances, where two $\mathbf{S}$-instances are in the same equivalence class precisely if they are homomorphically equivalent. We then have the first required property for a metric, namely, that the distance between two equivalence classes is 0 if and only if they are the same equivalence class. We now discuss the other three properties: nonnegativity, symmetry and triangle inequality. Clearly, d is nonnegative and symmetric. As for the triangle inequality, it is easy to see that d in fact satisfies the following strengthened version of the triangle inequality: $d(I_1, I_3) \leq \max\{d(I_1, I_2), d(I_2, I_3)\}$ (this makes d not just a metric, but an *ultrametric*). This metric d has been studied extensively in graph theory, where it has been used to characterize *restricted dualities* (see [NdM09] for a survey).

Returning to Example 3.6, the first statement implies that C_{2i+1}, $i \geq 1$, is a *Cauchy* sequence, that is, for every $\epsilon > 0$, there is a positive integer n such that if $i, j \geq n$, then $d(C_{2i+1}, C_{2j+1}) < \epsilon$. The second statement shows that $\lim_{i \to \infty} C_{2i+1} = K_2$ (this fact has been pointed out in [NdM09]). It is easy to see that a limit, when it exists, is unique up to homomorphic equivalence. The third statement implies that K_i, $i \geq 1$, is also a Cauchy sequence. However, there is *no* finite graph H such that $\lim_{i \to \infty} K_i = H$. This is so because if m is the size of the biggest clique contained in some finite graph H, then for all $i > |H|$, we have that $\mathrm{sim}(K_i, H) \leq m + 1$, hence $d(K_i, H) \geq 1/(m + 1)$. This shows that d is not a *complete* metric space. The completion of d (obtained by adding limits of all Cauchy sequences–the same way that the real numbers are obtained from the rational numbers) plays an important role in the characterization of restricted dualities [NdM09].

A function $\mathcal{F}$ is *uniformly continuous* if for every $\epsilon > 0$, there is $\delta > 0$ such that if $d(I_1, I_2) < \delta$, then $d(\mathcal{F}(I_1), \mathcal{F}(I_2)) < \epsilon$.[2] It is easy to see that under our definitions, a function $\mathcal{F}$ from the class of $\mathbf{S}$-instances to the class of $\mathbf{T}$-instances is CQ-local if and only if it is uniformly continuous. This helps demonstrate the naturalness of the notion of CQ-locality.

We shall show that for GLAV mappings, the chase is CQ-local. We shall show this result by two different proofs. The first proof makes use of earlier results in the literature, but gives very large bounds on the size of N (a stack of exponentials in n) The second proof is direct, and gives a bound on N that is polynomial in n.

We now consider D-locality for another choice of D. We first give another definition.

DEFINITION 3.7. Assume that I_1 and I_2 are $\mathbf{S}$-instances over a schema $\mathbf{S}$, and let n be a positive integer. We say that I_1 and I_2 are *FO$_n$-equivalent*, and write $I_1 \equiv_n I_2$, if I_1 and I_2 satisfy the same first-order formulas of quantifier depth at most n. □

The binary relations $\equiv_n$, $n \geq 1$, give rise to the family

$$\text{FO} \quad = \quad \{\equiv_n : n \geq 1\}.$$

Thus, $\mathcal{F}$ is FO-local if for every positive integer n, there is a positive integer N such that if $I_1 \equiv_N I_2$, then $\mathcal{F}(I_1) \equiv_n \mathcal{F}(I_2)$.

This notion of FO-locality was used, but not named, in the full, unpublished version of [ABFL04], to help prove non-rewritability of queries in data exchange. FO-locality is somewhat similar to the notion in [ABFL04] of "local consistency under FO-equivalence".

3.1 CQ-Local vs. FO-Local

In what follows, we will explore the relationship between CQ-local transformations and FO-local transformations. In doing so, we shall make use of the following theorem, which follows from the proof of Theorem 1.9 of Rossman [Ros08], and which is the main technical tool in proving his deep theorem on preservation under homomorphisms in the finite.

THEOREM 3.8 ([ROS08]). *Assume that $\mathbf{S}$ is a schema. For every positive integer n, there is a positive integer N and a function f_n such that for all $\mathbf{S}$-instances I_1 and I_2, the following hold:*

1. *$I_1 \leftrightarrow f_n(I_1)$ and $I_2 \leftrightarrow f_n(I_2)$.*
2. *If $I_1 \equiv_N^{cq} I_2$, then $f_n(I_1) \equiv_n f_n(I_2)$.*

We say that $\mathcal{F}$ *preserves homomorphic equivalence* if whenever $I_1 \leftrightarrow I_2$, then $\mathcal{F}(I_1) \leftrightarrow \mathcal{F}(I_2)$. Note that this is yet another notion of D-locality; indeed, $\mathcal{F}$ preserves homomorphism equivalence precisely when $\mathcal{F}$ is H-local, where $\text{H} = \{H_n : n \geq 1\}$ and $\text{H}_n = \leftrightarrow$, for every $n \geq 1$.

For example, if $\mathbf{M}$ is a GLAV mapping, then the chase procedure w.r.t. $\mathbf{M}$ preserves homomorphic equivalence; that is, if $I_1 \leftrightarrow I_2$, then $chase_{\mathbf{M}}(I_1) \leftrightarrow chase_{\mathbf{M}}(I_2)$ This is so because, as shown in [FKMP05], if $I_1 \rightarrow I_2$, then $chase_{\mathbf{M}}(I_1) \rightarrow chase_{\mathbf{M}}(I_2)$. In fact, the same holds true for schema mappings specified by SO tgds, as well as for schema mappings specified by a finite set of s-t tgds and a finite set of full target tgds. Note, however, that if target egds or arbitrary target tgds are allowed in the specification of a schema mapping $\mathbf{M}$, then the chase procedure need not be a total function, that is, $chase_{\mathbf{M}}(I)$ may not exist for some source instance I.

PROPOSITION 3.9. *If $\mathcal{F}$ is CQ-local, then $\mathcal{F}$ preserves homomorphic equivalence.*

[2]This is *uniform* continuity, since δ does not depend on the choice of I_1 or I_2. If the choice of δ depended on I_1, then we would have continuity of $\mathcal{F}$ at I_1, rather than uniform continuity of $\mathcal{F}$.

PROOF. Assume that $\mathcal{F}$ is CQ-local, and that $I_1 \leftrightarrow I_2$; we must show that $\mathcal{F}(I_1) \leftrightarrow \mathcal{F}(I_2)$. Let n be the maximum of the number of members of the active domains of $\mathcal{F}(I_1)$ and $\mathcal{F}(I_2)$. It is easy to see that $\mathcal{F}(I_1) \equiv_n^{cq} \mathcal{F}(I_2)$ if and only if $\mathcal{F}(I_1) \leftrightarrow \mathcal{F}(I_2)$. Since $\mathcal{F}$ is CQ-local, there is N such that if $I_1 \equiv_N^{cq} I_2$, then $\mathcal{F}(I_1) \equiv_n^{cq} \mathcal{F}(I_2)$. Since $I_1 \leftrightarrow I_2$, we have $I_1 \equiv_N^{cq} I_2$, and so $\mathcal{F}(I_1) \equiv_n^{cq} \mathcal{F}(I_2)$, hence $\mathcal{F}(I_1) \leftrightarrow \mathcal{F}(I_2)$, as desired. □

As we shall see in Fact 3.11, the converse of Proposition 3.9 fails. We now make use of Rossman's Theorem (Theorem 3.8) to prove the next result.

THEOREM 3.10. *If $\mathcal{F}$ preserves homomorphic equivalence and is* FO*-local, then $\mathcal{F}$ is* CQ*-local.*

PROOF. Assume that $\mathcal{F}$ is FO-local. Given the positive integer n, let n' be the positive integer guaranteed by FO-locality of $\mathcal{F}$, such that whenever $I_1 \equiv_{n'} I_2$, then $\mathcal{F}(I_1) \equiv_n \mathcal{F}(I_2)$. Let N be the positive integer guaranteed by Theorem 3.8 when the role of n is played by n'.

Assume now that $I_1 \equiv_N^{cq} I_2$; we must show that $\mathcal{F}(I_1) \equiv_n^{cq} \mathcal{F}(I_2)$. Let $f_{n'}$ be as in Theorem 3.8 when the role of n is played by n'. Since $I_1 \equiv_N^{cq} I_2$, it follows from Theorem 3.8 that $f_{n'}(I_1) \equiv_{n'} f_{n'}(I_2)$. It therefore follows from our choice of n' and by FO-locality that $\mathcal{F}(f_{n'}(I_1)) \equiv_n \mathcal{F}(f_{n'}(I_2))$. Since Boolean conjunctive queries with at most n variables are a special case of first-order formulas with quantifier depth at most n, it follows that

$$\mathcal{F}(f_{n'}(I_1)) \equiv_n^{cq} \mathcal{F}(f_{n'}(I_2)). \tag{2}$$

Now $I_1 \leftrightarrow f_{n'}(I_1)$ by Theorem 3.8 when the role of n is played by n'. Since $\mathcal{F}$ preserves homomorphic equivalence, it follows that

$$\mathcal{F}(I_1) \leftrightarrow \mathcal{F}(f_{n'}(I_1)). \tag{3}$$

Similarly,

$$\mathcal{F}(I_2) \leftrightarrow \mathcal{F}(f_{n'}(I_2)). \tag{4}$$

By Lemma 3.3, it follows from (2), (3), and (4) that $\mathcal{F}(I_1) \equiv_n^{cq} \mathcal{F}(I_2)$, as desired. □

FACT 3.11. As we now discuss, neither assumption in Theorem 3.10 can be dropped. First, the assumption that $\mathcal{F}$ is FO-local is needed. That is, there is $\mathcal{F}$ that preserves homomorphic equivalence, but is not CQ-local (so the converse of Proposition 3.9 fails). Here is the reason. If $\mathbf{M}$ is a schema mapping specified by a finite set of s-t tgds and full target tgds, then as we noted, the chase with $\mathbf{M}$ preserves homomorphic equivalence. However, such a chase need not be CQ-local, as we shall show in Theorem 3.20.

We now show that the assumption in Theorem 3.10 that $\mathcal{F}$ preserves homomorphic equivalence is needed. That is, there is $\mathcal{F}$ that is FO-local, but is not CQ-local. Let $\mathcal{F}$ be a function that maps every graph with at least two nodes (where a *node* is a member of the active domain) to a triangle (a cycle of length 3), and every graph with one node to a single edge. It is easy to see that $\mathcal{F}$ is FO-local – in fact, we can always take $N = 2$, since if $I_1 \equiv_2 I_2$, then I_1 has at least two nodes if and only if I_2 has at least two nodes, and so if $I_1 \equiv_2 I_2$, then $\mathcal{F}(I_1)$ and $\mathcal{F}(I_2)$ are isomorphic.

To show that $\mathcal{F}$ is not CQ-local, we need only show (by Proposition 3.9) that $\mathcal{F}$ does not preserve homomorphic equivalence. Let I_1 consist of a single node with a self-loop, and let I_2 consist of two nodes, each with a self-loop. It is easy to see that $I_1 \leftrightarrow I_2$. However, $\mathcal{F}(I_1)$ is a single edge, and $\mathcal{F}(I_2)$ is a triangle, and so $\mathcal{F}(I_1) \not\leftrightarrow \mathcal{F}(I_2)$.

The next proposition states what we have just shown.

PROPOSITION 3.12. *There is a transformation $\mathcal{F}$ that is* FO*-local but not* CQ*-local.*

We now show that the converse of Theorem 3.10 fails. While it is true that CQ-locality implies preservation of homomorphic equivalence (Proposition 3.9), the next proposition says that CQ-locality does not imply FO-locality.

PROPOSITION 3.13. *There is a transformation $\mathcal{F}$ that is* CQ*-local but not* FO*-local.*

PROOF. Define $\mathcal{F}_{\text{core}}$ by letting $\mathcal{F}_{\text{core}}(I)$ be the core of I. We now show that $\mathcal{F}_{\text{core}}$ is CQ-local, where we let $N = n$. Thus, assume that $I_1 \equiv_n^{cq} I_2$; we must show that $\mathcal{F}_{\text{core}}(I_1) \equiv_n^{cq} \mathcal{F}_{\text{core}}(I_2)$. Now $\mathcal{F}_{\text{core}}(I_1) \leftrightarrow I_1$, and $\mathcal{F}_{\text{core}}(I_2) \leftrightarrow I_2$. Since also $I_1 \equiv_n^{cq} I_2$, it follows from Lemma 3.3 that $\mathcal{F}_{\text{core}}(I_1) \equiv_n^{cq} \mathcal{F}_{\text{core}}(I_2)$.

We now show that $\mathcal{F}_{\text{core}}$ is not FO-local. Assume that it is; we shall derive a contradiction. Since by assumption $\mathcal{F}_{\text{core}}$ is FO-local, there is N such that if $I_1 \equiv_N I_2$, then $\mathcal{F}_{\text{core}}(I_1) \equiv_2 \mathcal{F}_{\text{core}}(I_2)$ (thus, we are taking $n = 2$, and finding N corresponding to n). It is well known that given N, there is N' such that if I_1 and I_2 are each undirected cycles with at least N' nodes, then $I_1 \equiv_N I_2$ (this follows, for example, from Theorem 4.3 of [FSV95]). Take I_1 to be an odd undirected cycle with at least N' nodes, and I_2 to be an even undirected cycle with at least N' nodes, It is straightforward to verify that $\mathcal{F}_{\text{core}}(I_1) = I_1$, and $\mathcal{F}_{\text{core}}(I_2)$ consists of a single edge of I_2. It follows easily that $\mathcal{F}_{\text{core}}(I_1) \not\equiv_2 \mathcal{F}_{\text{core}}(I_2)$. This is our desired contradiction. □

We feel that Propositions 3.12 and 3.13, along with Theorem 3.10, show an interesting relationship between two notions of locality: FO-locality and CQ-locality. We proved Theorem 3.10 using Rossman's Theorem. We do not know whether there is a proof of Theorem 3.10 that does not require the depth of Rossman's Theorem.

3.2 CQ-Locality of the Chase Procedure for GLAV Mappings

To give our first proof of the CQ-locality of the chase for GLAV mappings, we make use of the following theorem, which is a special case of a result in the full, unpublished version of [ABFL04].

THEOREM 3.14. *Let $\mathbf{M}$ be a GLAV mapping. Then the chase with respect to $\mathbf{M}$ is* FO*-local.*

Our first proof that the chase with respect to GLAV mappings is CQ-local follows immediately by combining Theorem 3.10 with Proposition 3.9 and Theorem 3.14.

THEOREM 3.15. *Let $\mathbf{M}$ be a GLAV mapping. Then the chase with respect to $\mathbf{M}$ is* CQ*-local.*

We now show that Theorem 3.15 generalizes from schema mappings specified by a finite set of s-t tgds to schema mappings specified by a second-order tgd (SO tgd). It is shown in [FKPT05] that SO tgds have a chase procedure, that produces a universal solution.

COROLLARY 3.16. *If $\mathbf{M}$ is a schema mapping specified by an SO tgd, then the chase with respect to $\mathbf{M}$ is* CQ*-local.*

PROOF. We first show that the composition of CQ-local transformations is CQ-local. Assume that $\mathcal{F}_1$ and $\mathcal{F}_2$ are CQ-local, and let n be a positive integer. Since $\mathcal{F}_1$ is CQ-local, we know that there is n' such that if $\mathcal{F}_2(I_1) \equiv_{n'}^{cq} \mathcal{F}_2(I_2)$, then $\mathcal{F}_1(\mathcal{F}_2(I_1)) \equiv_n^{cq} \mathcal{F}_1(\mathcal{F}_2(I_2))$. Since $\mathcal{F}_2$ is CQ-local, there is N such that if $I_1 \equiv_N^{cq} I_2$, then $\mathcal{F}_2(I_1) \equiv_{n'}^{cq} \mathcal{F}_2(I_2)$. Therefore, if $I_1 \equiv_N^{cq} I_2$, then we have $\mathcal{F}_1(\mathcal{F}_2(I_1)) \equiv_n^{cq} \mathcal{F}_1(\mathcal{F}_2(I_2))$, and so $\mathcal{F}_1 \circ \mathcal{F}_2$ is CQ-local.

Since the schema mapping $\mathbf{M}$ is specified by an SO tgd, it follows from [AFN11] that there are GLAV mappings $\mathbf{M}_1$ and $\mathbf{M}_2$ such that $\mathbf{M} = \mathbf{M}_1 \circ \mathbf{M}_2$.[3] It is shown in [Fag07, Proposition 7.2] that $\mathit{chase}_{\mathbf{M}_2}(\mathit{chase}_{\mathbf{M}_1}(I))$ is a universal solution for I with respect to $\mathbf{M}_1 \circ \mathbf{M}_2$. Further, it is shown in [FKPT05, Theorem 6.8] that $\mathit{chase}_{\mathbf{M}}(I)$ is universal for I with respect to $\mathbf{M}$. Since $\mathbf{M} = \mathbf{M}_1 \circ \mathbf{M}_2$, and since all universal solutions are homomorphically equivalent, it follows that $\mathit{chase}_{\mathbf{M}}(I) \leftrightarrow \mathit{chase}_{\mathbf{M}_2}(\mathit{chase}_{\mathbf{M}_1}(I))$.

Since (1) the chase with respect to $\mathbf{M}_1$ and the chase with respect to $\mathbf{M}_2$ are each CQ-local (by Theorem 3.15), (2) the composition of CQ-local transformations is CQ-local, and (3) $\mathit{chase}_{\mathbf{M}}(I) \leftrightarrow \mathit{chase}_{\mathbf{M}_2}(\mathit{chase}_{\mathbf{M}_1}(I))$, it follows that the chase with respect to $\mathbf{M}$ is CQ-local, which was to be shown. □

Theorem 3.15 tells us that for every positive integer n, there is a positive integer $N(n)$ (that, in general, depends on n) such that if $I_1 \equiv_{N(n)}^{cq} I_2$, then $\mathit{chase}_{\mathbf{M}}(I_1) \equiv_n^{cq} \mathit{chase}_{\mathbf{M}}(I_2)$. The proof of Theorem 3.15 yields an $N(n)$ that is a stack of exponentials in n, because this blow-up occurs in the proof of Rossman's Theorem (Theorem 3.8) and, to date, no smaller bounds are known. In what follows, we give a direct proof of Theorem 3.15 with much improved bounds that does not make use of Rossman's Theorem. In fact, our direct proof gives $N(n)$ as a polynomial in n whose degree is equal to the maximum arity of the relation symbols in the target schema.

We begin by introducing a new family of binary relations between instances.

DEFINITION 3.17. Assume that I_1 and I_2 are $\mathbf{S}$-instances over a schema $\mathbf{S}$, and let n be a positive integer. We write $I_1 \rightarrow_n^{cq} I_2$ to denote that every Boolean conjunctive query with at most n variables that is true on I_1 is also true on I_2. □

Intuitively, $\rightarrow_n^{cq}$ is about preservation of conjunctive queries with at most n variables. As such, $\rightarrow_n^{cq}$ is a relaxation of $\equiv_n^{cq}$, since $I_1 \equiv_n^{cq} I_2$ if and only if $I_1 \rightarrow_n^{cq} I_2$ and $I_2 \rightarrow_n^{cq} I_1$. The binary relations $\rightarrow_n^{cq}$, $n \geq 1$, give rise to the family

$$\text{PCQ} \quad = \quad \{\rightarrow_n^{cq} : n \geq 1\},$$

where PCQ stands for "preservation of conjunctive queries".

The next theorem, which is the key step in our direct proof of Theorem 3.15, tells us that for GLAV mappings, the chase is PCQ-local, and also gives a polynomial bound on N.

THEOREM 3.18. *Assume that $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a GLAV mapping specified by a finite set Σ of s-t tgds.*

- *The chase with respect to $\mathbf{M}$ is* PCQ*-local.*
- *Let k be the number of relation symbols in the target schema $\mathbf{T}$, let r be the maximum arity of the relation symbols in $\mathbf{T}$, and let m be the maximum number of universally quantified variables in the s-t tgds in Σ. For every natural number n, let $N(n) = mkn^r$. If I_1, I_2 are source instances such that $I_1 \rightarrow_{N(n)}^{cq} I_2$, then $\mathit{chase}_{\mathbf{M}}(I_1) \rightarrow_n^{cq} \mathit{chase}_{\mathbf{M}}(I_2)$.*

PROOF. Let I_1 and I_2 be $\mathbf{S}$-instances such that $I_1 \rightarrow_{N(n)}^{cq} I_2$. Put $J_1 = \mathit{chase}_{\mathbf{M}}(I_1)$ and $J_2 = \mathit{chase}_{\mathbf{M}}(I_2)$. We have to show that $J_1 \rightarrow_n^{cq} J_2$. In order to show this, it suffices to show that if $\exists z_1 \cdots \exists z_n \theta(z_1, \ldots, z_n)$ is a Boolean conjunctive query such that $J_1 \models \exists z_1 \cdots \exists z_n \theta(z_1, \ldots, z_n)$, then we also have that $J_2 \models$

[3] Our proof of Corollary 3.16 could simply make use of the weaker fact, proved in [FKPT05], that a schema mapping $\mathbf{M}$ specified by an SO tgd is the composition of a finite number m of GLAV mappings, but we may as well use the stronger fact that we can take $m = 2$.

$\exists z_1 \cdots \exists z_n \theta(z_1, \ldots, z_n)$. Let $a_1, \ldots, a_n$ be (not necessarily distinct) elements from J_1 such that $J_1 \models \theta(a_1, \ldots, a_n)$. Note that $\theta(a_1, \ldots, a_n)$ can be viewed as a collection of facts from the $\mathbf{T}$-instance J_1. Since $\mathbf{T}$ has k relation symbols, each of which is of arity at most r, it follows that $\theta(a_1, \ldots, a_n)$ consists of at most kn^r distinct facts $f_1, \ldots, f_{kn^r}$ from J_1. Since $J_1 = \mathit{chase}_{\mathbf{M}}(I_1)$, it follows that for each such fact f_j, where $1 \leq j \leq kn^r$, there are an s-t tgd $\forall \mathbf{x}_j(\varphi_j(\mathbf{x}_j) \rightarrow \exists \mathbf{y}_j \psi_j(\mathbf{x}_j, \mathbf{y}_j))$ in Σ, a tuple $\mathbf{c}_j$ of elements from I_1, and a tuple $\mathbf{d}_j$ of elements from J_1 such that

- $I_1 \models \varphi_j(\mathbf{c}_j)$;
- $J_1 \models \psi_j(\mathbf{c}_j, \mathbf{d}_j)$;
- f_j is one of the facts occurring in $\psi_j(\mathbf{c}_j, \mathbf{d}_j)$.

By renaming variables as needed, we may assume that the tuples $\mathbf{x}_j$ and $\mathbf{x}_{j'}$ have no variables in common if $j \neq j'$ (for $1 \leq j \leq kn^r$ and $1 \leq j' \leq kn^r$). Since every s-t tgd in Σ has at most m universally quantified variables, it follows that the total number of variables in $\mathbf{x}_1, \ldots, \mathbf{x}_{kn^r}$ is at most mkn^r. Note that each a_i is either a null in J_1 that is not in I_1 or it is equal to an element occurring in at least one tuple $\mathbf{c}_j$. Furthermore, if it is a null in J_1 that is not in I_1, then a_i is the witness to one and only one existentially quantified variable in one of the above s-t tgds from Σ. Note also that two tuples $\mathbf{c}_j$ and $\mathbf{c}_l$ may have elements in common. Let $\chi(\mathbf{x}_1, \ldots, \mathbf{x}_{kn^r})$ be a conjunction of equalities such that $\chi(\mathbf{c}_1, \ldots, \mathbf{c}_{kn^r})$ is a complete list of all equalities that hold between the elements from I_1 that occur in the tuples $\mathbf{c}_1, \ldots, \mathbf{c}_{kn^r}$. Consequently,

$$I_1 \models \exists \mathbf{x}_1 \cdots \exists \mathbf{x}_{kn^r}((\bigwedge_{j=1}^{kn^r} \varphi_j(\mathbf{x}_j)) \wedge \chi(\mathbf{x}_1, \ldots, \mathbf{x}_{kn^r})).$$

Note that the formula in the preceding expression is logically equivalent to a conjunctive query with (at most) $N(n) = mkn^r$ variables. Since $I_1 \rightarrow^{cq}_{N(n)} I_2$, we have that

$$I_2 \models \exists \mathbf{x}_1 \cdots \exists \mathbf{x}_{kn^r}((\bigwedge_{j=1}^{kn^r} \varphi_j(\mathbf{x}_j)) \wedge \chi(\mathbf{x}_1, \ldots, \mathbf{x}_{kn^r})).$$

Let $\mathbf{c}'_1, \ldots, \mathbf{c}'_{kn^r}$ be tuples of elements from I_2 such that $I_2 \models (\bigwedge_{j=1}^{kn^r} \varphi_j(\mathbf{c}'_j)) \wedge \chi(\mathbf{c}'_1, \ldots, \mathbf{c}'_{kn^r})$. As a result of the chase procedure, there are tuples $\mathbf{d}'_1, \ldots, \mathbf{d}'_{kn^r}$ from J_2 such that $J_2 \models \bigwedge_{j=1}^{kn^r} \psi_j(\mathbf{c}'_j, \mathbf{d}'_j)$.

We will show that $J_2 \models \exists z_1 \cdots \exists z_n \theta(z_1, \ldots, z_n)$. In fact, we will show that the existential quantifiers $\exists z_i$, $1 \leq i \leq n$, in this conjunctive query can be witnessed by elements b_i, $1 \leq i \leq n$, chosen from the tuples $\mathbf{c}'_1, \ldots, \mathbf{c}'_{kn^r}, \mathbf{d}'_1, \ldots, \mathbf{d}'_{kn^r}$ in a way that we now describe. For $i \leq n$, let a_i be the element that witnessed the existential quantifier $\exists z_i$ in J_1. We distinguish two cases.

Case 1: The element a_i is a null in J_1 that is not in I_1. In this case, every occurrence of a_i in the facts $f_1, \ldots, f_{kn^r}$ arises from only one tuple $\mathbf{c}_j$ and from only one s-t tgd $\forall \mathbf{x}_j(\varphi_j(\mathbf{x}_j) \rightarrow \exists \mathbf{y}_j \psi_j(\mathbf{x}_j, \mathbf{y}_j))$ in Σ; moreover, a_i witnesses one and only one existential quantifier, say $\exists y$, in the tuple $\exists \mathbf{y}_j$. In this case, we take b_i to be the element from the tuple $\mathbf{d}'_j$ that witnesses $\exists y$ in J_2. Note that b_i is a null in J_2 that is not in I_2.

Case 2: The element a_i is in I_1. In this case, a_i may occur in several different tuples $\mathbf{c}_j$. Pick one of them, say $\mathbf{c}_r$. Let b_i be the element of I_2 that occurs in the tuple $\mathbf{c}'_r$ and in the same position as a_i does in $\mathbf{c}_r$. Note that b_i is an element of J_2. Moreover, if a tuple $\mathbf{c}_s$ different from $\mathbf{c}_r$ had been chosen where a_i occurs in $\mathbf{c}_s$, then the same element b_i would have been obtained.

Since J_1 satisfying $\bigwedge_{j=1}^{kn^r} \psi_j(\mathbf{c}_j, \mathbf{d}_j)$ has the effect that J_1 satisfies $\theta(a_1, \ldots, a_n)$, and since J_2 satisfies $\bigwedge_{j=1}^{kn^r} \psi_j(\mathbf{c}'_j, \mathbf{d}'_j)$, this tells us (from our mimicking construction, where b_i mimics a_i) that J_2 satisfies $\theta(b_1, \ldots, b_n)$. Hence, $J_2 \models \exists z_1 \cdots \exists z_n \theta(z_1, \ldots, z_n)$, as desired. □

As an immediate consequence of Theorem 3.18, we obtain a significantly improved version of Theorem 3.15 in which N has a polynomial dependence on n. In fact, the degree of the polynomial is equal to the maximum arity of the target schema.

THEOREM 3.19. (Theorem 3.15 revisited) *Assume that $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a GLAV mapping specified by a finite set Σ of s-t tgds. Let k be the number of relation symbols in the target schema $\mathbf{T}$, let r be the maximum arity of the relation symbols in $\mathbf{T}$, and let m be the maximum number of universally quantified variables in the s-t tgds in Σ. For every natural number n, let $N(n) = mkn^r$. If I_1 and I_2 are source instances such that $I_1 \equiv^{cq}_{N(n)} I_2$, then $\mathit{chase}_{\mathbf{M}}(I_1) \equiv^{cq}_{n} \mathit{chase}_{\mathbf{M}}(I_2)$.*

3.3 Failures of CQ-Locality

The next theorem says that Theorem 3.15 cannot be extended to allow target dependencies.

THEOREM 3.20. *There is a schema mapping $\mathbf{M}$ specified by three s-t tgds and by a full target tgd such that the chase with respect to $\mathbf{M}$ is not* CQ*-local.*

PROOF. Define the schema mapping $\mathbf{M}$ as follows. The source schema consists of a binary relation symbol P, and unary relation symbols R and S. The target schema consist of a binary relation symbol P', and unary relation symbols R' and S'. The dependencies of $\mathbf{M}$ are:

$$P(x, y) \rightarrow P'(x, y), \quad R(x) \rightarrow R'(x), \quad S(x) \rightarrow S'(x),$$
$$P'(x, y) \wedge P'(y, z) \rightarrow P'(x, z).$$

In the full version of this paper, we show that $\mathbf{M}$ is not CQ-local. □

In Corollary 3.16, we showed that if $\mathbf{M}$ is a schema mapping specified by an SO tgd, then the chase with respect to $\mathbf{M}$ is CQ-local. However, as a corollary of Theorem 3.20, we now show that this is not true when the schema mapping is specified by an st-SO dependency, as defined in [AFN11]. These st-SO dependencies are similar to SO tgds, but allow equalities in the conclusion. A notion of the chase for st-SO dependencies, which produces a universal solution, is defined in [AFN11].

COROLLARY 3.21. *There is a schema mapping $\mathbf{M}$ specified by an st-SO dependency such that the chase with respect to $\mathbf{M}$ is not* CQ*-local.*

PROOF. Let $\mathbf{M}$ be as in the proof of Theorem 3.20, where the chase with respect to $\mathbf{M}$ is not CQ-local. Let $\mathbf{M}'$ be the "copy" schema mapping specified by the s-t tgds $P'(x, y) \rightarrow P''(x, y)$, $R'(x) \rightarrow R''(x)$, and $S'(x) \rightarrow S''(x)$. Let $\mathbf{M}'' = \mathbf{M} \circ \mathbf{M}'$. Then $\mathbf{M}''$ is the same as $\mathbf{M}$, up to a renaming of relation symbols. It is shown in [AFN11] that if $\mathbf{M}_1$ is a schema mapping specified by s-t tgds, target egds, and a weakly acyclic [FKMP05] set of target tgds, and $\mathbf{M}_2$ is a schema mapping specified by s-t tgds, then $\mathbf{M}_1 \circ \mathbf{M}_2$ is a schema mapping specified by an st-SO dependency. Therefore, $\mathbf{M}''$ is specified by an st-SO dependency. The result of the chase using the st-SO dependency that specifies $\mathbf{M}''$ is a universal solution w.r.t. d$\mathbf{M}''$. But the universal solutions for $\mathbf{M}''$ are the same as

the universal solutions for $\mathbf{M}$, up to a renaming of relation symbols, since $\mathbf{M}''$ is the same as $\mathbf{M}$, up to a renaming of relation symbols. Since the chase with respect to $\mathbf{M}$ is not CQ-local, it follows easily that the chase with respect to $\mathbf{M}''$ is not CQ-local. □

4. Degrees of Equivalence of Schema Mappings

Schema mappings $\mathbf{M}_1$ and $\mathbf{M}_2$ are CQ-*equivalent* [FKNP08] if $certain_{\mathbf{M}_1}(q, I) = certain_{\mathbf{M}_2}(q, I)$ for every (not necessarily Boolean) conjunctive query q and every source instance I. As mentioned in the Introduction, Madhavan and Halevy [MH03] based their notion of composition on CQ-equivalence. Later on, Fagin et al. [FKNP08] studied CQ-equivalence in the context of schema mapping optimization, while Arenas et al. [APRR09] studied CQ-equivalence in the context of inverting schema mappings.

The two main questions we will focus on in this section are:

- When is the composition of two GLAV mappings logically equivalent to a GLAV mapping?
- When is the composition of two GLAV mappings CQ-equivalent to a GLAV mapping?

The way we deal with these problems is to divide GLAV mappings into a small number of well-studied classes, namely GAV, LAV, extended LAV, and general GLAV (of course, these classes are not mutually exclusive), and see when the composition of schema mappings from these various classes can be guaranteed to be a GLAV mapping, and also to see when they can be guaranteed to be CQ-equivalent to a GLAV mapping. It turns out that up to now, there has been one gap in each of these scenarios, and we will fill both of these gaps, in order to obtain a complete picture.

We also consider a bounded form of CQ-equivalence. If n is a positive integer and $\mathbf{M}_1$, $\mathbf{M}_2$ are two schema mappings, then $\mathbf{M}_1$ and $\mathbf{M}_2$ are CQ_n-*equivalent* if $certain_{\mathbf{M}_1}(q, I) = certain_{\mathbf{M}_2}(q, I)$ for every (not necessarily Boolean) conjunctive query q with at most n variables and for every source instance I. It follows from [MH03, Proposition 3] that for every $n \geq 1$, the composition of two GLAV mappings is always CQ_n-equivalent to some GLAV mapping. We feel that it is useful to give a direct proof of this fact, which we do in Section 4.3.

4.1 Logical Equivalence

In the case of logical equivalence, the gap in our knowledge until now has been the question as to whether the composition of a GLAV mapping with a LAV mapping is necessarily logically equivalent to a GLAV mapping. Our next theorem answers this positively. This generalizes a result of Arocena, Fuxman, and Miller [AFM10], that the composition of LAV mappings is logically equivalent to a GLAV mapping (in fact, to a LAV mapping). Our proof also provides an alternative proof that the composition of LAV mappings is logically equivalent to a LAV mapping, since in our proof that a GLAV mapping composed with a LAV mapping is GLAV, it happens that if the first mapping is LAV, then the composition is actually specified by LAV constraints.

We begin with a lemma. Recall that an *element*, or *value*, is an entry of a tuple of a relation. If f is a function on the elements, and I is an instance, then we write $f(I)$ for the result of replacing every element x in every tuple of I by $f(x)$.

LEMMA 4.1. *Let $\mathbf{M}$ be a LAV mapping. If J is a solution for I with respect to $\mathbf{M}$, and f is an arbitrary function on the elements, then $f(J)$ is a solution for $f(I)$ with respect to $\mathbf{M}$.*

PROOF. This follows fairly easily from the viewpoint that f is simply a renaming of elements (not necessarily one-to-one), and LAV tgds are indifferent to renamings (thus, they fire in the same way on tuples whether or not some entries of the tuple are equal). The feature of LAV tgds that we used in this argument is that no variable appears twice in a premise. □

We now define a *restriction* of a tgd $\alpha \rightarrow \exists\bar{y}\beta$. Let X be the set of variables that appear in α, let X' be a subset of X, and let F be a function from X to X' that maps every variable in X' into itself. Let $\alpha' \rightarrow \exists\bar{y}\beta'$ be the result of modifying $\alpha \rightarrow \exists\bar{y}\beta$ by replacing every variable x in X by $F(x)$. Then we call the tgd $\alpha' \rightarrow \exists\bar{y}\beta'$ a *restriction* of the tgd $\alpha \rightarrow \exists\bar{y}\beta$. For example, the tgd $R(x, x, z, x) \rightarrow \exists y Q(x, x, y)$ is a restriction of the tgd $R(w, x, z, w) \rightarrow \exists y Q(w, x, y)$, where w is mapped to x.

THEOREM 4.2. *If $\mathbf{M}_{12}$ is a GLAV mapping and $\mathbf{M}_{23}$ is a LAV mapping, then $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is logically equivalent to a GLAV mapping.*

PROOF. *(Sketch)* Let $\mathbf{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathbf{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where Σ_{12} is a finite set of s-t tgds, and Σ_{23} is a finite set of LAV s-t tgds. For convenience, we assume that Σ_{12} is closed under restriction (this is without loss of generality, since a tgd logically implies each of its restrictions). We now define Σ_{13}, and we shall show that for the schema mapping $\mathbf{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$, we have $\mathbf{M}_{13} = \mathbf{M}_{12} \circ \mathbf{M}_{23}$. Our definition of Σ_{13} is different from that given in [AFM10]. We describe our construction of Σ_{13} somewhat informally, by speaking about chasing formulas to get other formulas. For each tgd $\alpha \rightarrow \exists\bar{y}\beta$ in Σ_{12}, we chase β with Σ_{23}, call the result δ, and let $\alpha \rightarrow \exists\bar{q}\delta$ be a member of Σ_{13}, where $\bar{q}$ consists of the variables in δ but not α.

We now show that Σ_{13} specifies the composition. We first show that if $(I_1, I_2) \models \Sigma_{12}$ and $(I_2, I_3) \models \Sigma_{23}$, then $(I_1, I_3) \models \Sigma_{13}$. This is immediate, since the result of a chase is "forced". We conclude by showing that if $(I_1, I_3) \models \Sigma_{13}$, then there is I_2 such that $(I_1, I_2) \models \Sigma_{12}$ and $(I_2, I_3) \models \Sigma_{23}$. To simplify the discussion, assume without loss of generality that I_1 contains only constants.

Let us define a *restricted chase* of an instance I to be one where a tgd $\alpha \rightarrow \exists\bar{y}\beta$ is applied only when there is a *one-to-one* homomorphism of α into I. Since by assumption, Σ_{12} is closed under restriction, it follows easily that a restricted chase of I is a universal solution for I with respect to Σ. Let J_2 be the result of doing a restricted chase of I_1 with Σ_{12}. We shall discuss how to assign a value $f(n)$ (which may be a constant or a null) to each null n in J_2 to obtain I_2. Assume that the tgd $\alpha \rightarrow \exists\bar{y}\beta$ fires in the restricted chase of I_1 with Σ_{12}. Let J_2' be the subset of J_2 that is obtained by one firing of this tgd. We now define a function f that assigns values to the nulls of J_2'. The s-t tgd $\alpha \rightarrow \exists\bar{y}\beta$ yields J_2' in the restricted chase of I_1 with Σ_{12} because of a one-to-one homomorphism h from α to I_1. Since h is one-to-one, the relation corresponding to the formula β in our identification of formulas with instances is J_2', up to a one-to-one renaming of variables by values. Let $\alpha \rightarrow \exists\bar{q}\delta$ be the member of Σ_{13} that arises from the tgd $\alpha \rightarrow \exists\bar{y}\beta$ in our construction of Σ_{13}. Since $(I_1, I_3) \models \alpha \rightarrow \exists\bar{q}\delta$, it follows from our construction of $\alpha \rightarrow \exists\bar{q}\delta$ that there is a homomorphism h' from U, the result of chasing J_2' with Σ_{23}, into I_3, where h' respects I_1 (maps constants into themselves). Define f to agree with h' on the active domain of U, and to be the identity otherwise. Since U is a solution for J_2' with respect to Σ_{23}, it follows from Lemma 4.1 that $f(U)$ is a solution for $f(J_2')$ with respect to Σ_{23}. That is, $(f(J_2'), f(U)) \models \Sigma_{23}$. Since $f(U) = h'(U) \subseteq I_3$, it follows that $(f(J_2'), I_3) \models \Sigma_{23}$.

If a different J_2' (call it J_2'') arises from a different step of the restricted chase, then the active domains of J_2' and J_2'' have in common at most constants, on which f is the identity. So if we repeat

this process to define $f(n)$ for every null n in J_2, we obtain a well-defined function f. Let $I_2 = f(J_2)$. Since (1) $(f(J_2'), I_3) \models \Sigma_{23}$ for each J_2' in our construction, (2) I_2 is the union of these instances $f(J_2')$, and (3) Σ_{23} is extended LAV (all we need for this argument is that the premises of the s-t tgds are singletons), it follows that $(I_2, I_3) \models \Sigma_{23}$. Furthermore, $(I_1, I_2) \models \Sigma_{12}$, since J_2 is a solution for I_1 (it is even universal), and I_2 is a homomorphic image of J_2 under a homomorphism (namely, f) that respects I_1 (and the solutions of I_1 w.r.t. Σ_{12} are closed under homomorphisms that respect I_1). Since we have shown that $(I_1, I_2) \models \Sigma_{12}$ and $(I_2, I_3) \models \Sigma_{23}$, this completes the proof. □

COROLLARY 4.3 ([AFM10]). *If both $\mathbf{M}_{12}$ and $\mathbf{M}_{23}$ are LAV mappings, then $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is logically equivalent to a LAV mapping.*

PROOF. In the construction of the composition formula Σ_{13} in the proof of Theorem 4.2, each premise of Σ_{13} is a premise of Σ_{12}. So if Σ_{12} is LAV, then so is Σ_{13}. □

Let us now consider Table 1, about the results of composition. When an entry under the "Logical Equivalence" column is GLAV, this means that the composition is guaranteed to be logically equivalent to a GLAV mapping. For example, the entry under "Logical Equivalence" for the row GAV ∘ GLAV says "GLAV", and this means that the composition of a GAV mapping with a GLAV mapping is always logically equivalent to a GLAV mapping. When an entry is "Not GLAV", this means that there is an example where that composition is not logically equivalent to any GLAV mapping. For example, the entry under "Logical Equivalence" for the row LAV ∘ ex. LAV says "Not GLAV", and this means that there is a LAV mapping $\mathbf{M}_{12}$ and an extended LAV mapping $\mathbf{M}_{23}$ such that $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is not logically equivalent to any GLAV mapping.

If we now look at the first two columns ("Composition" and "Logical Equivalence") of Table 1, it is straightforward to verify that we have covered all combinations of composing LAV, extended LAV, GAV, and GLAV up to logical equivalence (that is, they are easily inferred from what is in the table). For example, the case of GAV ∘ extended LAV is covered by the case of GAV ∘ GLAV, in the sense that because GAV ∘ GLAV is necessarily logically equivalent to a GLAV mapping, so is GAV ∘ extended LAV. As another example, the case of extended LAV ∘ extended LAV is covered by the case of LAV ∘ extended LAV, in the sense that because there is an example of LAV ∘ extended LAV where the result is not logically equivalent to any GLAV mapping, this negative example covers also extended LAV ∘ extended LAV.

4.2 CQ-equivalence

Let us consider an example from [FKPT05]. There are three schemas $\mathbf{S}_1$, $\mathbf{S}_2$ and $\mathbf{S}_3$. Schema $\mathbf{S}_1$ consists of a single unary relation symbol $\mathtt{Emp}$ of employees. Schema $\mathbf{S}_2$ consists of a single binary relation symbol $\mathtt{Mgr}_1$, that associates each employee with a manager. Schema $\mathbf{S}_3$ consists of a similar binary relation symbol $\mathtt{Mgr}$, that is intended to provide a copy of $\mathtt{Mgr}_1$. and an additional unary relation symbol $\mathtt{SelfMgr}$, that is intended to store employees who are their own manager. Consider now the schema mappings $\mathbf{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathbf{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where

$$\Sigma_{12} = \{\, \forall e\, (\mathtt{Emp}(e) \rightarrow \exists m \mathtt{Mgr}_1(e, m)) \,\}$$

$$\Sigma_{23} = \{\, \forall e \forall m\, (\mathtt{Mgr}_1(e, m) \rightarrow \mathtt{Mgr}(e, m)),\ \forall e(\mathtt{Mgr}_1(e, e) \rightarrow \mathtt{SelfMgr}(e)) \,\}.$$

The SO tgd that specifies the composition $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is given in (1) in Section 2. It is shown in [FKPT05] that this SO tgd is not logically equivalent to any finite (or even infinite) set of s-t tgds. Note that $\mathbf{M}_{12}$ is LAV, and $\mathbf{M}_{23}$ is extended LAV. Can we say anything positive about the composition of a LAV mapping with an extended LAV mapping? In Theorem 4.6, we show that such a composition (and even more, the composition of an arbitrary GLAV mapping with an extended LAV mapping) is always CQ-equivalent to a GLAV mapping. The proof of this theorem depends on a characterization in [FKNP08] about when a schema mapping specified by an SO tgd is CQ-equivalent to a GLAV mapping. We begin with some definitions from [FKNP08].

DEFINITION 4.4. Assume that $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a schema mapping, where Σ is either a finite set of s-t tgds or an SO tgd.

- Let I be a source instance and K a target instance.

  The *Gaifman graph of facts of K w.r.t. I* is a graph whose nodes are the facts of K, and with an edge between two facts if they have in common some element not in the active domain of I.[4]

  A *fact block* (or simply *f-block*) of K w.r.t. I is a connected component of the Gaifman graph of facts of K w.r.t. I.

  The *f-block size* of K w.r.t. I is the maximum size of the f-blocks of K w.r.t. I.

  When I is understood from the context, we simply refer to the Gaifman graph of facts of K, the f-blocks of K, and the f-block size of K.

- We say that $\mathbf{M}$ (or Σ) has *bounded f-block size* if there is a positive integer b such that for every source instance I, the f-block size of $\mathrm{core}(\mathit{chase}_{\mathbf{M}}(I))$ w.r.t. I is at most b. We then refer to the minimal such b as the *f-block size of* $\mathbf{M}$ (or *of* Σ). □

We have the following theorem from [FKNP08].

THEOREM 4.5 ([FKNP08]). *A schema mapping $\mathbf{M}$ specified by an SO tgd is CQ-equivalent to a schema mapping specified by a finite set of s-t tgds if and only if $\mathbf{M}$ has bounded f-block size.*

We can now prove that the composition of a GLAV mapping with an extended LAV mapping is CQ-equivalent to a GLAV mapping.

THEOREM 4.6. *If $\mathbf{M}_{12}$ is a GLAV mapping and $\mathbf{M}_{23}$ is an extended LAV mapping, then $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is CQ-equivalent to a GLAV mapping.*

PROOF. Let $\mathbf{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathbf{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ be schema mappings, where Σ_{12} is a finite sets of s-t tgds, and Σ_{23} is a finite set of extended LAV constraints. Let $\mathbf{M}_{13} = \mathbf{M}_{12} \circ \mathbf{M}_{23}$. We must show that $\mathbf{M}_{13}$ is CQ-equivalent to a GLAV mapping.

By [FKPT05], we know that there is an SO tgd Σ_{13} such that $\mathbf{M}_{13} = (\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$. It follows from Theorem 4.5 that to prove the theorem, it is sufficient to show that $\mathrm{core}(\mathit{chase}_{\Sigma_{13}}(I))$ has bounded f-block size w.r.t. I. Since, as shown in the proof of Corollary 3.16, $\mathit{chase}_{\Sigma_{13}}(I)$ and $\mathit{chase}_{\Sigma_{23}}(\mathit{chase}_{\Sigma_{12}}(I))$ are homomorphically equivalent, and since homomorphically equivalent instances have the same core (up to isomorphism), it is sufficient for us to show that $\mathrm{core}(\mathit{chase}_{\Sigma_{23}}(\mathit{chase}_{\Sigma_{12}}(I)))$ has bounded f-block size.w.r.t. I. So it suffices to show that the f-blocks of $\mathit{chase}_{\Sigma_{23}}(\mathit{chase}_{\Sigma_{12}}(I))$ have sizes bounded by a constant that depends only on Σ_{12} and Σ_{23} (this is so because the core of an instance K is a subinstance of K, hence a bound on the sizes of the f-blocks of K is inherited by the core of K).

[4] In [FKNP08], it was assumed that the source instance I consists only of constants, and so the Gaifman graph of K was defined not w.r.t. I, but instead by defining the Gaifman graph of facts of K to be a graph whose nodes are the facts of K, and with an edge between two facts if they have a null value in common,

Let n be the maximum number of atoms in the conclusions of the s-t tgds in Σ_{12}, let m be the maximum number of atoms in the conclusions of the s-t tgds in Σ_{23}, and let s be the number of s-t tgds in Σ_{23}. Let I be an $\mathbf{S}_1$-instance. We claim that every f-block of $\mathit{chase}_{\Sigma_{23}}(\mathit{chase}_{\Sigma_{12}}(I))$ is of size at most nms. To see this, first note that every f-block of $\mathit{chase}_{\Sigma_{12}}(I)$ is of size at most n. Fix now an s-t tgd, say τ, in Σ_{23}. Since τ is an extended LAV s-t tgd (that is, it has a singleton premise), when we chase $\mathit{chase}_{\Sigma_{12}}(I)$ with Σ_{23}, we produce f-blocks that have size at most nm. By going over all tgds in Σ_{23}, we have that the f-blocks of $\mathit{chase}_{\Sigma_{23}}(\mathit{chase}_{\Sigma_{12}}(I))$ are of size at most nms. □

The preceding result enables us to complete the picture on CQ-equivalence, as given in the third column ("CQ-equivalence") of Table 1. For this CQ-equivalence column, just as for the Logical Equivalence column, it is straightforward to verify that we have covered all combinations of composing LAV, extended LAV, GAV, and GLAV up to CQ-equivalence (that is, they are easily inferred from what is in the table). The first three entries in the CQ-equivalence column of Table 1 (those with no citation) follow immediately from the corresponding entries in the Logical Equivalence column of the table. The fourth and fifth entries in the CQ-equivalence column of Table 1 follow from Theorem 4.6.

4.3 Bounded CQ-Equivalence

Again, let us begin with an example from [FKPT05]. 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 $\texttt{Takes}_1$, 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 $\mathbf{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathbf{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$, where

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

It is shown in [FKPT05] that the composition $\mathbf{M}_{12} \circ \mathbf{M}_{23}$ is not CQ-equivalent to any GLAV mapping. Note that $\mathbf{M}_{12}$ is LAV, and $\mathbf{M}_{23}$ is GAV. Can we say anything positive about the composition of a LAV mapping with a GAV mapping? The next proposition says that in fact the composition of any pair of GLAV mappings is always CQ_n-equivalent to a GLAV mapping. As we noted, this result follows from [MH03, Proposition 3]. We feel that it is useful to give a direct proof of this result, which we now do. In the fourth column ("CQ_n-equivalence) of Table 1, all entries are GLAV, which follows since the last entry is GLAV.

PROPOSITION 4.7. *Let n be a positive integer. The composition of GLAV mappings is CQ_n-equivalent to a GLAV mapping.*

PROOF. Let $\mathbf{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathbf{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ be schema mappings, where Σ_{12} and Σ_{23} are finite sets of s-t tgds. By [FKPT05], we know that there is an SO tgd σ that specifies $\mathbf{M}_{12} \circ \mathbf{M}_{23}$. Let I be an $\mathbf{S}_1$-instance, and let J be a result of chasing I with σ. Let $c_1, \ldots, c_r$ be the distinct elements of I, and let $d_1, \ldots, d_m$ be the distinct remaining elements of J. Let ϕ_I be the formula that is the conjunction of all atoms over $x_1, \ldots, x_r$ that hold in I when x_i plays the role of c_i, for each i. For example, if $R(c_3, c_7)$ holds in I, then one conjunct is $R(x_3, x_7)$. Let ψ_I be the formula that is the conjunction of all atoms over $x_1, \ldots, x_r, y_1, \ldots, y_m$ 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 $S(c_3, d_9)$ holds in J, then one conjunct is $S(x_3, y_9)$. Let τ_I be the s-t tgd

$$\forall x_1 \cdots \forall x_r(\phi_I \rightarrow \exists y_1 \cdots \exists y_m \psi_I).$$

Intuitively, τ_I describes exactly a result of chasing I with σ.

Let n be as in the statement of the proposition, let d be the number of relation symbols in $\mathbf{S}_3$, let r be the maximum arity of a relation symbol of $\mathbf{S}_3$, and let $N = dn^r$. It is easy to see that each conjunctive query over $\mathbf{S}_3$ with at most n variables has at most N distinct atoms. Let k be the maximum number of atoms in a premise of a conjunct ("tgd part") of σ. Let Σ be the set of s-t tgds $\tau_{I'}$ where I' has at most Nk facts. Let q be an arbitrary conjunctive query over $\mathbf{S}_3$ with at most n variables, and let I be a source instance. By definition of CQ_n-equivalence, we need only show that $\mathit{certain}_{\Sigma}(q, I) = \mathit{certain}_{\sigma}(q, I)$. Now $\mathit{certain}_{\Sigma}(q, I) \subseteq \mathit{certain}_{\sigma}(q, I)$, since Σ is a logical consequence of σ. We now show the opposite inclusion. Assume that $\bar{e} \in \mathit{certain}_{\sigma}(q, I)$; we wish to show that $\bar{e} \in \mathit{certain}_{\Sigma}(q, I)$. Let J be a result of chasing I with σ. So $\bar{e} \in q(J)$. Since q has at most N atoms, there is J' with at most N facts such that $J' \subseteq J$ and $\bar{e} \in q(J')$. There is then I_0 with at most Nk facts such that J' is in the result of chasing I_0 with σ. Let J_0 be the result of chasing I_0 with σ. Since $J' \subseteq J_0$, it follows that $\bar{e} \in q(J_0)$. Let J_1 be the result of chasing I with Σ. So J_1 contains the result of chasing I_0 with Σ, which contains the result of chasing I_0 with τ_{I_0}, which contains J_0. We just showed that $J_0 \subseteq J_1$. Since $\bar{e} \in q(J_0)$, it then follows that $\bar{e} \in q(J_1)$. Hence, since J_1 is a universal solution for I with respect to Σ, it follows from Theorem 2.2 that $\bar{e} \in \mathit{certain}_{\Sigma}(q, I)$, as desired. □

5. Deciding CQ-equivalence

Let $\mathbf{M}_1$ and $\mathbf{M}_2$ be two given schema mappings, each specified by either a finite set of s-t tgds or by an SO tgd. Assume that we wish to tell whether $\mathbf{M}_1$ and $\mathbf{M}_2$ are logically equivalent, and also whether they are CQ-equivalent. For each of these two decision problems, there are three cases to consider.

1. $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *are both GLAV:* It follows from Proposition 3.14 of [FKNP08] that two such mappings are CQ-equivalent if and only if they are logically equivalent. Moreover, telling whether two given finite sets of s-t tgds are logically equivalent is a decidable problem, by using the chase [ABU79, MMS79].
2. $\mathbf{M}_1$ *and* $\mathbf{M}_2$ *are both specified by SO tgds:* Telling whether two given SO tgds are logically equivalent is an undecidable problem [FPSS11, Theorem 1]. The decidability status of telling whether two given SO tgds are CQ-equivalent is not known. However, as also shown in [FPSS11], this problem does become undecidable in the presence of additional source key constraints, that is, in the case where $\mathbf{M_1}$ and $\mathbf{M_2}$ are each specified by an SO tgd and a finite set of source key constraints.
3. *One of* $\mathbf{M}_1$ *or* $\mathbf{M}_2$ *is specified by an SO tgd, and the other is GLAV:* Telling whether a given SO tgd and a given finite set of s-t tgds are logically equivalent is an undecidable problem. This follows by examining the proof of Theorem 1 in [FPSS11], which actually is derived from an undecidability result in [APR09] about inverses of schema mappings. In contrast, here we show that telling whether a given SO tgd and a given finite set of s-t tgds are CQ-equivalent is a decidable problem.

Table 1: Results of composition

Composition	Logical Equivalence	CQ-Equivalence	CQ_n-Equivalence
GAV ∘ GAV	GLAV (even GAV) [FKPT05]	GLAV (even GAV)	GLAV (even GAV)
GAV ∘ GLAV	GLAV [FKPT05]	GLAV	GLAV
GLAV ∘ LAV	GLAV Theorem 4.2; [AFM10] for LAV ∘ LAV	GLAV	GLAV
LAV ∘ ex. LAV	Not GLAV [FKPT05]	GLAV Theorem 4.6	GLAV
GLAV ∘ ex. LAV	Not GLAV [FKPT05]	GLAV Theorem 4.6	GLAV
LAV ∘ GAV	Not GLAV [FKPT05]	Not GLAV [FKPT05]	GLAV
GLAV ∘ GLAV	Not GLAV [FKPT05]	Not GLAV [FKPT05]	GLAV [MH03]; Proposition 4.7

As the first step in showing our decidability result, we prove the next proposition. An *f-block* is defined in Definition 4.4.

PROPOSITION 5.1. *The following two decision problems are reducible to each other.*

- *Given an SO tgd σ and a finite set Σ of s-t tgds, is σ CQ-equivalent to Σ?*
- *Given an SO tgd σ and a positive integer b, is the f-block size of σ bounded by b?*

PROOF. The proof of Theorem 4.10 in [FKNP08] shows that, given an SO tgd σ and a positive integer b, we can construct a finite set $\Sigma_{\sigma,b}$ of s-t tgds with the property: the f-block size of σ is bounded by b if and only if σ is CQ-equivalent to $\Sigma_{\sigma,b}$.[5]

We now show that the first problem is reducible to the second. Suppose we are given an SO tgd σ and a finite set Σ of s-t tgds, and we want to test whether or not σ is CQ-equivalent to Σ. Let b be the maximum number of atoms in the conclusions of the tgds in Σ; as pointed out in [FKNP08], the f-block size of Σ is bounded by b. We first test whether or not the f-block size of σ is bounded by b. If the answer is "no", then σ is not CQ-equivalent to Σ. This is because if σ and Σ were CQ-equivalent, then it follows from Theorem 3.5 of [FKNP08] that for each source instance I, necessarily $\mathrm{core}(\mathit{chase}_\sigma(I))$ and $\mathrm{core}(\mathit{chase}_\Sigma(I))$ would be isomorphic, and so the f-block sizes of σ and Σ would be the same. So assume that the answer is 'yes". By our earlier comment, it follows that σ is CQ-equivalent to $\Sigma_{\sigma,b}$. So σ is CQ-equivalent to Σ if and only if Σ is CQ-equivalent to $\Sigma_{\sigma,b}$. As we noted earlier, it follows from Proposition 3.14 of [FKNP08] that for finite sets of s-t tgds, logical equivalence coincides with CQ-equivalence. So Σ is CQ-equivalent to $\Sigma_{\sigma,b}$ if and only if Σ is logically equivalent to $\Sigma_{\sigma,b}$. But it is decidable whether Σ is logically equivalent to $\Sigma_{\sigma,b}$, by using the chase [ABU79, MMS79].

Next we show that the second problem is reducible to the first. For this, given an SO tgd σ and a bound b, we first construct the set $\Sigma_{\sigma,b}$ and then test whether or not σ is CQ-equivalent to $\Sigma_{\sigma,b}$. As we noted, σ is CQ-equivalent to $\Sigma_{\sigma,b}$ if and only if the f-block size of σ is bounded by b. □

We now prove the decidability of the question in the second bullet of Proposition 5.1.

[5]This property of $\Sigma_{\sigma,b}$ is not stated explicitly in that proof, but it can be derived from that proof by in particular noting that the f-block size of $\Sigma_{\sigma,b}$ is bounded by b. We remark that $\Sigma_{\sigma,b}$ consists of s-t tgds of the form τ_I, as defined in Proposition 4.7.

PROPOSITION 5.2. *There is an algorithm for the following problem: Given an SO tgd σ and a positive integer b, is the f-block size of σ bounded by b?*

PROOF. Let σ be an SO tgd, and let $r(\sigma)$ be the maximum number of atoms in any of the premises inside of σ. It is shown in the proof of Proposition 4.8 in [FKNP08] that $r(\sigma)$ witnesses that σ has *bounded support*, that is to say, for every source instance I and every target instance J, if $J \xrightarrow{I} \mathrm{core}(\mathit{chase}_\sigma(I))$, then there is a subinstance I' of I such that $|I'| \leq r(\sigma)|J|$ and $J \xrightarrow{I'} \mathrm{core}(\mathit{chase}_\sigma(I'))$. [6]

Consider the following algorithm: given an SO tgd σ and a positive integer b, go over all source instances I' such that $|I'| \leq r(\sigma)(b+1)$ (there are only finitely many such instances, and they can be computed from σ and b). For each such instance I', compute the f-block size of $\mathrm{core}(\mathit{chase}_\sigma(I'))$. If one of these f-block sizes is bigger than b, report that the f-block size of σ is bigger than b; otherwise, report that the f-block size of σ is at most b.

For the correctness of the algorithm, it is clear that if one of the computed f-block sizes is bigger than b, then the f-block size of σ is greater than b. For the other direction, we will show that if the f-block size of σ is bigger than b, then there is an instance I' such that $|I'| \leq r(\sigma)(b+1)$ and the f-block size of $\mathrm{core}(\mathit{chase}_\sigma(I'))$ is bigger than b. So, assume that the f-block size of σ is bigger than b. Then there is a source instance I such that the f-block size of $\mathrm{core}(\mathit{chase}_\sigma(I))$ is at least $b+1$. Consider an f-block C of $\mathrm{core}(\mathit{chase}_\sigma(I))$ of size at least $b+1$. Let J be a subset of C such that $|J| = b+1$ and J is a connected subgraph of the Gaifman graph of facts of $\mathrm{core}(\mathit{chase}_\sigma(I))$ w.r.t. I. Since $J \subseteq \mathrm{core}(\mathit{chase}_\sigma(I))$, we have $J \xrightarrow{I} \mathrm{core}(\mathit{chase}_\sigma(I))$, and so by our earlier comments, there is a subinstance I' of I such that $|I'| \leq r(\sigma)|J| = r(\sigma)(b+1)$ and $J \xrightarrow{I} \mathrm{core}(\mathit{chase}_\sigma(I'))$. Let h be the homomorphism from J to $\mathrm{core}(\mathit{chase}_\sigma(I'))$ that respects I. Since $I' \subseteq I$, it follows that h is a homomorphism from J to $\mathrm{core}(\mathit{chase}_\sigma(I))$. Therefore, since J is a part of an f-block in $\mathrm{core}(\mathit{chase}_\sigma(I))$, it follows that h cannot map J into anything smaller than J, so h simply renames the nulls in a one-to-one manner. Moreover, the image of J under this homomorphism h is a connected subgraph of the Gaifman graph of facts

[6]In [FKNP08] it was assumed that I consists only of constants, and that homomorphisms map each constant onto itself, and so $\rightarrow$ rather than $\xrightarrow{I}$ was used in the definition of bounded support.

of $\mathrm{core}(\mathit{chase}_\sigma(I'))$, since the facts of J form a connected graph. Hence, $\mathrm{core}(\mathit{chase}_\sigma(I'))$ contains a f-block of size at least $b+1$, which was to be shown. □

By combining Proposition 5.1 with Proposition 5.2, we obtain the following result.

THEOREM 5.3. *There is an algorithm for the following decision problem: Given a schema mapping $\mathbf{M_1}$ specified by an SO tgd and a GLAV mapping $\mathbf{M_2}$, is $\mathbf{M_1}$ CQ-equivalent to $\mathbf{M_2}$?*

COROLLARY 5.4. *There is an algorithm for the following decision problem: Given three GLAV mappings $\mathbf{M_1}$, $\mathbf{M_2}$, and $\mathbf{M_3}$, is $\mathbf{M_1} \circ \mathbf{M_2}$ CQ-equivalent to $\mathbf{M_3}$?*

PROOF. In [FKPT05], there is algorithm for finding an SO tgd σ that is logically equivalent to $\mathbf{M_1} \circ \mathbf{M_2}$. We then check whether σ is CQ-equivalent to $\mathbf{M_3}$, by making use of the algorithm guaranteed by Theorem 5.3. □

Madhavan and Halevy [MH03] claim without proof that the decision problem in Corollary 5.4 is in Π_2^p. This claim would imply Theorem 5.3, since it is shown in [AFN11] that given a schema mapping $\mathbf{M}$ specified by an SO tgd, there is a procedure for finding GLAV mappings $\mathbf{M}_1$ and $\mathbf{M}_2$ such that $\mathbf{M} = \mathbf{M}_1 \circ \mathbf{M}_2$.

6. Concluding Remarks

We have introduced the notion of a CQ-local transformation. Intuitively, a CQ-local transformation has the property that if two instances are indistinguishable using conjunctive queries of a sufficiently large size N, then their images under the transformation are also indistinguishable using conjunctive queries of a given size n. We proved that for GLAV mappings, the chase is CQ-local, and showed that N can be taken to be polynomial in n. One way of looking at the CQ-locality of the chase is that the chase is "uniformly continuous". We showed that if target dependencies are allowed, then CQ-locality of the chase may fail.

We investigated several different notions of equivalence of schema mappings and completed the picture as to when the composition of schema mappings from various subclasses of GLAV mappings is guaranteed to be logically equivalent to a GLAV mapping, and when it is guaranteed to be CQ-equivalent to a GLAV mapping.

Finally, we proved that the following problem is decidable: given an SO tgd and a finite set of s-t tgds, are they CQ-equivalent? This result sheds light on the differences between CQ-equivalence and logical equivalence, since the following problem is known to be undecidable: given an SO tgd and a finite set of s-t tgds, are they logically equivalent?

There are several interesting issues to pursue. A concrete technical question is the decidability of the following problem: given an SO tgd, is it CQ-equivalent to *some* GLAV mapping? It follows from results in [AFN11] that this is equivalent to the decidability of the following problem: given two GLAV mappings, is their composition CQ-equivalent to *some* GLAV mapping? More broadly, we feel that the notion of CQ-locality (and its alternate interpretation as uniform continuity) is potentially a valuable tool, with much more to be explored.

7. References

[ABFL04] M. Arenas, P. Barceló, R. Fagin, and L. Libkin. Locally consistent transformations and query answering in data exchange. In *ACM Symp. on Principles of Database Systems*, pages 229–240, 2004.

[ABLM10] M. Arenas, P. Barceló, L. Libkin, and F. Murlak. *Relational and XML Data Exchange*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2010.

[ABU79] A. V. Aho, C. Beeri, and J. D. Ullman. The theory of joins in relational databases. *ACM Trans. on Database Systems*, 4(3):297–314, 1979.

[AFM10] P.C. Arocena, A. Fuxman, and R.J. Miller. Composing local-as-view mappings: closure and applications. In *Int. Conf. on Database Theory*, pages 209–218, 2010.

[AFN11] M. Arenas, R. Fagin, and A. Nash. Composition with target constraints. *Logical Methods in Computer Science*, 7(3:13):1–38, 2011.

[APR09] M. Arenas, J. Pérez, and C. Riveros. The recovery of a schema mapping: Bringing exchanged data back. *ACM Trans. on Database Systems*, 34(4), 2009.

[APRR09] M. Arenas, J. Pérez, J.L. Reutter, and C. Riveros. Inverting schema mappings: Bridging the gap between theory and practice. *PVLDB*, 2(1):1018–1029, 2009.

[CM77] A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *ACM Symp. on Theory of Computing*, pages 77–90, 1977.

[Fag07] R. Fagin. Inverting schema mappings. *ACM Trans. on Database Systems*, 32(4), 2007.

[FKMP05] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: Semantics and query answering. *Theoretical Computer Science*, 336(1):89–124, 2005.

[FKNP08] R. Fagin, P. G. Kolaitis, A. Nash, and L. Popa. Towards a theory of schema-mapping optimization. In *ACM Symp. on Principles of Database Systems*, pages 33–42, 2008.

[FKPT05] R. Fagin, P. G. Kolaitis, L. Popa, and W.-C. Tan. Composing schema mappings: Second-order dependencies to the rescue. *ACM Trans. on Database Systems*, 30(4):994–1055, 2005.

[FKPT11] R. Fagin, P.G. Kolaitis, L. Popa, and W-C. Tan. Schema mapping evolution through composition and inversion. In Z. Bellahsene, A. Bonifati, and E. Rahm, editors, *Schema Matching and Mapping*, pages 191–222. Springer, 2011.

[FPSS11] I. Feinerer, R. Pichler, E. Sallinger, and V. Savenkov. On the undecidability of the equivalence of second-order tuple generating dependencies. In *Alberto Mendelzon Workshop*, 2011.

[FSV95] R. Fagin, L. Stockmeyer, and M. Y. Vardi. On monadic NP vs. monadic co-NP. *Inf. and Computation*, 120(1):78–92, July 1995.

[HN92] P. Hell and J. Nešetřil. The core of a graph. *Discrete Mathematics*, 109:117–126, 1992.

[Kol05] P. G. Kolaitis. Schema mappings, data exchange, and metadata management. In *ACM Symp. on Principles of Database Systems*, pages 61–75, 2005.

[Len02] M. Lenzerini. Data integration: A theoretical perspective. In *ACM Symp. on Principles of Database Systems*, pages 233–246, 2002.

[MH03] J. Madhavan and A. Y. Halevy. Composing mappings among data sources. In *Int. Conf. on Very Large Data Bases*, pages 572–583, 2003.

[MMS79] D. Maier, A. O. Mendelzon, and Y. Sagiv. Testing implications of data dependencies. *ACM Trans. on Database Systems*, 4(4):455–469, 1979.

[NdM09] J. Nešetřil and P. Ossona de Mendez. From sparse graphs to nowhere dense structures: Decompositions, independence, dualities and limits. In *Proc. of the Fifth European Congress of Mathematics*, 2009.

[PSS11] R. Pichler, E. Sallinger, and V. Savenkov. Relaxed notions of schema mapping equivalence revisited. In *Int. Conf. on Database Theory*, pages 90–101, 2011.

[Ros08] B. Rossman. Homomorphism preservation theorems. *J. ACM*, 55(3), 2008.