Reverse Data Exchange: Coping with Nulls

RONALD FAGIN, IBM Research - Almaden
PHOKION G. KOLAITIS, UC Santa Cruz and IBM Research - Almaden
LUCIAN POPA, IBM Research - Almaden
WANG-CHIEW TAN, IBM Research - Almaden and UC Santa Cruz

An inverse of a schema mapping $\mathcal{M}$ is intended to undo what $\mathcal{M}$ does, thus providing a way to perform reverse data exchange. In recent years, three different formalizations of this concept have been introduced and studied, namely the notions of an inverse of a schema mapping, a quasi-inverse of a schema mapping, and a maximum recovery of a schema mapping. The study of these notions has been carried out in the context in which source instances are restricted to consist entirely of constants, while target instances may contain both constants and labeled nulls. This restriction on source instances is crucial for obtaining some of the main technical results about these three notions, but, at the same time, limits their usefulness, since reverse data exchange naturally leads to source instances that may contain both constants and labeled nulls.

We develop a new framework for reverse data exchange that supports source instances that may contain nulls, and we thereby overcome the semantic mismatch between source and target instances of the previous formalizations. The development of this new framework requires a careful reformulation of all the important notions, including the notions of the identity schema mapping, inverse, and maximum recovery. To this effect, we introduce the notions of extended identity schema mapping, extended inverse, and maximum extended recovery, by making systematic use of the homomorphism relation on instances. We give results concerning the existence of extended inverses and of maximum extended recoveries, and results concerning their applications to reverse data exchange and query answering. Moreover, we show that maximum extended recoveries can be used to capture in a quantitative way, the amount of information loss embodied in a schema mapping specified by source-to-target tuple-generating dependencies.

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

Additional Key Words and Phrases: Schema mapping, data exchange, data integration, model management, inverse, quasi-inverse, maximum recovery, chase

ACM Reference Format:
Fagin, R., Kolaitis, P. G., Popa, L., and Tan, W.-C. 2011. Reverse data exchange: Coping with nulls. ACM Trans. Datab. Syst. 36, 2, Article 11 (May 2011), 42 pages.
DOI = 10.1145/1966385.1966389 http://doi.acm.org/10.1145/1966385.1966389

1. INTRODUCTION

Background and Motivation. Schema mappings are high-level specifications of how data from a source schema is to be transformed to data in a different schema, called

This article is an invited paper from the 2009 *ACM Symposium on Principles of Database Systems* (PODS).
P. G. Kolaitis is supported by NSF Grant IIS-0905276; W.-C. Tan is supported NSF Grant IIS-0905276 and NSF CAREER award IIS-0347065.

Author's address: R. Fagin email: fagin@almaden.ibm.com; P. G. Kolaitis email: kolaitis@cs.ucsc.edu; L. Popa email: lucian@almaden.ibm.com; W.-C. Tan email: wangchiew@us.ibm.com.

DOI 10.1145/1966385.1966389 http://doi.acm.org/10.1145/1966385.1966389

the target schema. More formally, a schema mapping is a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\mathbf{S}$ is a source schema, $\mathbf{T}$ is a target schema, and Σ is a set of database dependencies that specify the relationship between $\mathbf{S}$ and $\mathbf{T}$. In recent years, an extensive investigation of schema mappings and of their uses in data exchange and data integration has been carried out. One particular direction of this investigation has focused on the study of operators on schema mappings. Among all such operators originally introduced in Bernstein [2003], the composition operator and the inverse operator have been recognized as two fundamental ones. The intuition behind these two operators is as follows. Given two schema mappings, $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, such that the target schema of $\mathcal{M}_{12}$ is the same as the source schema of $\mathcal{M}_{23}$, the composition operator yields a schema mapping $\mathcal{M}_{13}$ that is equivalent to the successive application of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$, thus providing a way to perform data exchange directly between the source schema of $\mathcal{M}_{12}$ and the target schema of $\mathcal{M}_{23}$. Given a schema mapping $\mathcal{M}$, the inverse operator yields a schema mapping $\mathcal{M}'$ that undoes what $\mathcal{M}$ did, thus providing a way to do reverse data exchange. Clearly, these two operators are of interest in their own right; furthermore, when combined together, they attain even greater power, since in combination, they can be used to analyze schema evolution [Bernstein 2003; Fagin et al. 2011].

By now, the composition operator has been investigated in depth, and consensus has been achieved on what the definitive semantics for composition ought to be [Bernstein et al. 2008; Fagin et al. 2005b; Madhavan and Halevy 2003; Nash et al. 2005]. The state of affairs concerning the inverse operator, however, is more complicated and by far less definitive. In Fagin [2007], rigorous semantics for the inverse operator was given for the first time. The notion of inverse introduced in Fagin [2007] turned out to be rather restrictive, as most schema mappings specified by source-to-target tuple generating dependencies (s-t tgds) are not invertible. For this reason, the notion of a *quasi-inverse* of a schema mapping was introduced and studied in Fagin et al. [2008]. After this, a competing notion of a *maximum recovery* of a schema mapping was introduced and studied in Arenas et al. [2009]. For invertible schema mappings, the notions of inverse, quasi-inverse, and maximum recovery, coincide. In contrast, for noninvertible schema mappings, the notions of quasi-inverse and maximum recovery differ in general. Moreover, every schema mapping specified by a set of s-t tgds has a maximum recovery, whereas there are such schema mappings that are not quasi-invertible.

Their differences notwithstanding, all previous studies of inverse operators (inverse, quasi-inverse, and maximum recovery) have the following basic assumption in common. The source instances are *ground*, that is, they consist entirely of constants, while, on the contrary, target instances may contain both constants and labeled nulls (variables). In particular, some of the key technical results in Arenas et al. [2009]; Fagin [2007]; and Fagin et al. [2008] very much depend on the assumption that source instances do not contain labeled nulls. However, applications of inverse operators naturally lead to source instances that may contain labeled nulls, in addition to constants. The following example illustrates this scenario.

Example 1.1. Let $\mathcal{M}$ be the schema mapping specified by the tuple-generating dependency

$$P(x, y, z) \rightarrow Q(x, y) \wedge R(y, z),$$

which describes a decomposition of a source relation P into two target relations, Q and R. It was shown in Fagin et al. [2008] that $\mathcal{M}$ is not invertible but is quasi-invertible. Furthermore, a natural inverse of $\mathcal{M}$, which is both a quasi-inverse of $\mathcal{M}$ and a maximum recovery of $\mathcal{M}$ is the schema mapping $\mathcal{M}'$ specified by the following set of reverse tgds.

$$\Sigma' = \{Q(x, y) \rightarrow \exists z P(x, y, z),\ R(y, z) \rightarrow \exists x P(x, y, z)\}.$$

Consider the ground source instance $I = \{P(a, b, c)\}$, where a, b, c, are constants. The result of chasing I with $\mathcal{M}$ (the result of performing data exchange with $\mathcal{M}$) is the instance $U = \{Q(a, b), R(b, c)\}$. If we now chase U with $\mathcal{M}'$ (perform the reverse data exchange with $\mathcal{M}'$), we obtain the source instance $V = \{P(a, b, Z), P(X, b, c)\}$, where Z and X are nulls. Note that V, which is the canonical result of reverse data exchange, is no longer a ground instance, and thus it is ruled out from the semantics.

Another limitation of restricting source instances to be ground is that data exchange cannot be performed on source instances that are the result of a prior data exchange with s-t tgds, since in general, labeled nulls may be generated by chasing source instances with tgds to produce universal solutions [Fagin et al. 2005a]. Thus, the preceding considerations suggest that previous studies of inverse, quasi-inverse, and maximum recovery suffer from a semantic mismatch that stems from the assumption that only ground instances are allowed.

Summary of Contributions. Our goal in this article is to develop a new framework for reverse data exchange, which overcomes this semantic mismatch. This new framework supports source instances with nulls, and makes it possible to recover source instances using reverse data exchange and to permit target instances that result from one data exchange to be used as source instances of another data exchange (thus, in the long run, this framework will enable the analysis of schema evolution using composition and inverse). This framework is tailored mainly for schema mappings specified by finite sets of s-t tgds. However, the main definitions are stated for arbitrary schema mappings and some of the results are proved for this level of generality.

The development of this new framework requires a careful reformulation of all the important notions, including the notions of the identity schema mapping (which was used in the definition of inverse [Fagin 2007]), inverse, and maximum recovery. This is so because these earlier notions, which were studied under the assumption that only ground source instances are allowed, lose their desirable properties when schema mappings are used to perform data exchange between source and target instances that may both contain nulls. For example, although a key result of Arenas et al. [2009] is that every schema mapping specified by a finite set of s-t tgds has a maximum recovery (when only ground source instances are allowed), we show (Proposition 4.2) that the schema mapping specified by the tgd $P(x, y) \rightarrow \exists z(Q(x, z) \wedge Q(z, y))$ has no maximum recovery when source instances may contain nulls.

The key to finding the right notions in the new framework is to replace the containment relation $I_1 \subseteq I_2$ between instances by the homomorphism relation $I_1 \rightarrow I_2$, which holds if there is a homomorphism from I_1 to I_2 that maps constants to themselves (note that if I_1 is a ground instance, then $I_1 \rightarrow I_2$ if and only if $I_1 \subseteq I_2$). We use the notation $\rightarrow$ to denote the binary relation $\{(I_1, I_2) \mid I_1 \rightarrow I_2\}$ between instances that may contain nulls. The next step is to replace the identity schema mapping $\mathrm{Id} = \{(I_1, I_2) \mid I_1, I_2$ are ground instances such that $I_1 \subseteq I_2\}$ with the extended *identity* schema mapping $e(\mathrm{Id}) = \rightarrow$. Thus,

$$e(\mathrm{Id}) = \{(I_1, I_2) \mid I_1, I_2 \text{ are instances such that } I_1 \rightarrow I_2\}. \quad (1)$$

In fact, even the notion of a solution needs to be reformulated in the new framework. Specifically, we say that a target instance J is an *extended solution* for a source instance I with respect to a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ if there are a source instance I' and a target instance J' such that $I \rightarrow I'$, $(I', J') \models \Sigma$, and $J' \rightarrow J$. In effect, an extended solution J for I with respect to $\mathcal{M}$ is a solution for I with respect to the schema mapping $e(\mathcal{M}) = \rightarrow \circ \mathcal{M} \circ \rightarrow$, which we call the *homomorphic extension* of $\mathcal{M}$. (It is easy to verify that $e(\mathrm{Id})$ under this definition is exactly as given in (1).) The intuition behind extended solutions is that, since labeled nulls represent unknown information, it is legal to homomorphically map them to other values before or after taking the standard notion of solution. In a sense, the notion of an extended solution is a relaxation of the notion of solution, where the exact values of the labeled nulls (in both the source instance and the target instance) do not matter. Further, our definition of extended solutions relaxes the notion of solution even more by allowing the possibility that additional facts may be present (in the spirit of the "open-world assumption").[1] With these new notions at hand, we can then define the notions of *extended inverse*, *extended recovery*, and *maximum extended recovery* by systematically replacing the identity schema mapping Id by the extended identity schema mapping $e(\mathrm{Id})$, and the composition $\mathcal{M} \circ \mathcal{M}'$, by the composition $e(\mathcal{M}) \circ e(\mathcal{M}')$ of the homomorphic extensions of $\mathcal{M}$ and $\mathcal{M}'$. Specifically, we say that $\mathcal{M}'$ is an *extended inverse* of $\mathcal{M}$ if $e(\mathcal{M}) \circ e(\mathcal{M}') = e(\mathrm{Id})$. We say that $\mathcal{M}'$ is an *extended recovery* of $\mathcal{M}$ if for every source instance I, we have that $(I, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$; finally, we say that $\mathcal{M}'$ is a *maximum extended recovery* of $\mathcal{M}$ if $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$ and for every extended recovery $\mathcal{M}''$ of $\mathcal{M}$ we have that $e(\mathcal{M}) \circ e(\mathcal{M}') \subseteq e(\mathcal{M}) \circ e(\mathcal{M}'')$.

The first group of our technical results concerns the properties of the extended inverse and its relation to the inverse. We give a necessary and sufficient condition for an arbitrary schema mapping to have an extended inverse and then focus on schema mappings specified by s-t tgds. We show that if $\mathcal{M}$ is specified by s-t tgds and is extended invertible, then it is invertible, but not vice versa. An extended inverse, however, need not be an inverse (and, of course, an inverse need not be an extended inverse, since invertibility does not imply extended invertibility). We also show that if $\mathcal{M}$ and $\mathcal{M}'$ are schema mappings such that both are specified by tgds, then $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$ if and only if $\mathcal{M}'$ is a *chase-inverse* of $\mathcal{M}$, that is, if and only if every source instance I is homomorphically equivalent to $\mathit{chase}_{\mathcal{M}'}(\mathit{chase}_{\mathcal{M}}(I))$. (By $\mathit{chase}_{\mathcal{M}'}(\mathit{chase}_{\mathcal{M}}(I))$, we mean the source instance obtained from I by first chasing with $\mathcal{M}$ and then with $\mathcal{M}'$.) This result reveals that extended inverses specified by tgds are ideally suited for reverse data exchange, since if the original source instance I is no longer available, we can recover a homomorphically equivalent one by chasing a universal solution for I with an extended inverse specified by tgds. It should be noted that this result does not hold for non-extended inverses.

After this, we focus on maximum extended recoveries and obtain the main technical results of this article. We show that every schema mapping $\mathcal{M}$ specified by s-t tgds has a maximum extended recovery $\mathcal{M}'$. Note that if both $\mathcal{M}'$ and $\mathcal{M}''$ are maximum extended recoveries of $\mathcal{M}$, then

$$e(\mathcal{M}) \circ e(\mathcal{M}') = e(\mathcal{M}) \circ e(\mathcal{M}'').$$

For schema mappings $\mathcal{M}$ specified by s-t tgds, we characterize the quantity $e(\mathcal{M}) \circ e(\mathcal{M}')$ in terms of $\mathcal{M}$ alone by showing that $e(\mathcal{M}) \circ e(\mathcal{M}') = \rightarrow_{\mathcal{M}}$, where

$$\rightarrow_{\mathcal{M}} = \{(I_1, I_2) \mid \mathit{chase}_{\mathcal{M}}(I_1) \rightarrow \mathit{chase}_{\mathcal{M}}(I_2)\}.$$

This result makes it possible to capture in a quantitative way, the information loss embodied in a schema mapping. Specifically, we can define the information loss of a schema mapping $\mathcal{M}$ specified by s-t tgds as the set-theoretic difference $\rightarrow_{\mathcal{M}} \ \setminus e(\mathrm{Id})$; note also that $e(\mathrm{Id}) \ \subseteq \rightarrow_{\mathcal{M}}$. In effect, the information loss of a schema mapping $\mathcal{M}$ specified by s-t tgds measures the amount by which $\mathcal{M}$ deviates from being extended invertible, since we show that $\mathcal{M}$ is extended invertible if and only if $\rightarrow_{\mathcal{M}} = \ e(\mathrm{Id})$. The concept of information loss can also be used to compare two schema mappings in a quantitative way and determine which of the two is "more invertible."

[1] This is automatic in the case of schema mappings specified by dependencies, such as s-t tgds.

We also embark on an investigation of the language needed to express maximum extended recoveries of schema mappings specified by s-t tgds. For schema mappings that are specified by s-t tgds, we give a polynomial-time algorithm for producing a maximum extended recovery that is specified in existential second-order logic. It is an open problem as to whether every schema mapping that is specified by s-t tgds has a maximum extended recovery that can be specified in first-order logic.

Finally, we show that maximum extended recoveries specified by disjunctive tgds can be used to perform reverse data exchange and reverse query answering, where the latter means answering queries over the source schema when the source instance is no longer available. A key notion that is central to all these applications (reverse data exchange, reverse query answering, and also comparing schema mappings) is the notion of a disjunctive relaxed chase-inverse of a schema mapping, which provides a procedural counterpart to the notion of maximum extended recovery of a schema mapping.

Differences from Conference Version. A preliminary version of this article appeared in the proceedings of the 2009 ACM PODS conference [Fagin et al. 2009]. The differences between the present article and the conference version are as follows. Section 4.3, which gives an alternative notion of an extended recovery, is new. Section 4.4, which is also new, contains a discussion comparing our treatment of nulls with other work in the literature on incomplete databases. Theorem 5.1 in the conference version about a first-order language for maximum extended recoveries for schema mappings specified by full s-t tgds has been removed, because the proof was incorrect. However, Section 5 contains a new polynomial-time algorithm for producing a maximum extended recovery, specified in existential second-order logic, for a schema mapping specified by s-t tgds (not necessarily full). In addition, the present article contains a number of new examples, especially in Section 6. Finally, most of the proofs in this article did not appear in the conference version.

2. BACKGROUND, BASIC NOTIONS, AND NOTATION

A *schema* $\mathbf{R}$ is a finite sequence $\langle R_1, \ldots, R_k \rangle$ of relation symbols, each of a fixed arity. An *instance* I over $\mathbf{R}$ 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 $v_1, \ldots, v_m$ are values such that $(v_1, \ldots, v_m) \in R_i^I$. We shall often identify an instance with its set of facts.

We assume that we have a fixed countably infinite set $\underline{\mathsf{Const}}$ of *constants* and countably infinite set $\underline{\mathsf{Var}}$ of *labeled nulls* that is disjoint from $\underline{\mathsf{Const}}$. A *ground* instance over some schema is an instance such that all values occurring in its relations are constants. We will also assume that $\mathbf{S}$ is a fixed *source* schema and $\mathbf{T}$ is a fixed *target* schema. Starting with Fagin et al. [2005a], earlier work on data exchange was carried out under the assumption that instances over the source schema $\mathbf{S}$ are ground, while instances over the target schema $\mathbf{T}$ may contain both constants and labeled nulls. This models the situation in which we perform data exchange from $\mathbf{S}$ to $\mathbf{T}$ under the assumption that the individual values of source instances are known, while the underspecification of a data exchange setting may give rise to null values in the target instances. In this article, we will drop the assumption that source instances are ground; instead, we will assume that instances over both the source and the target schema may have individual values from $\underline{\mathsf{Const}} \cup \underline{\mathsf{Var}}$. By allowing source instances to contain both constants and labeled nulls, we model the situation where source instances may also contain incomplete information. In particular, under this assumption, the target instance of one data exchange can be used as source instances of another data exchange.

Next, we define two crucial concepts, homomorphism and homomorphic equivalence, that we use frequently throughout this article.

Definition 2.1. Let I_1 and I_2 be instances over a schema $\mathbf{R}$, where the values of I_1 and I_2 are drawn from $\underline{\mathsf{Const}} \cup \underline{\mathsf{Var}}$. A function h from $\underline{\mathsf{Const}} \cup \underline{\mathsf{Var}}$ to $\underline{\mathsf{Const}} \cup \underline{\mathsf{Var}}$ is a *homomorphism* from I_1 to I_2 if for every c in $\underline{\mathsf{Const}}$, we have that $h(c) = c$, and for every relation symbol R in $\mathbf{R}$ and every tuple $(a_1, \ldots, a_n) \in R^{I_1}$, we have that $(h(a_1), \ldots, h(a_n)) \in R^{I_2}$.

We use the notation $I_1 \to I_2$ to denote that there is a homomorphism from I_1 to I_2. We say that I_1 is *homomorphically equivalent* to I_2 if $I_1 \to I_2$ and $I_2 \to I_1$. Let $\to_{\mathbf{R}}$ denote the binary relation $\{(I_1, I_2) \mid I_1$ and I_2 are instances over $\mathbf{R}$ and $I_1 \to I_2\}$. In what follows and for simplicity of notation, we will use $\to$ instead of $\to_{\mathbf{R}}$ because the schema $\mathbf{R}$ will be understood from the context.

Schema mappings. A *schema mapping* is a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ is a set of constraints (typically, formulas in some logic) that describe the relationship between $\mathbf{S}$ and $\mathbf{T}$. We say that Σ *specifies* $\mathcal{M}$. This is the syntactic view of schema mappings. For all practical purposes, however, a schema mapping can be identified with the binary relation:

$$\{(I, J) \mid I \text{ is an } \mathbf{S}\text{-instance}, J \text{ is a } \mathbf{T}\text{-instance}, (I, J) \models \Sigma\}.$$

This is the semantic view of schema mappings. In what follows, we will switch freely between the syntactic view and the semantic view of schema mappings. In particular, we will use the notation $(I, J) \in \mathcal{M}$ to denote that the ordered pair (I, J) satisfies the constraints of $\mathcal{M}$; furthermore, we will sometimes define schema mappings by simply defining the set of ordered pairs (I, J) that constitute $\mathcal{M}$ (instead of giving a set of constraints that specify $\mathcal{M}$). Note that $\to$ is itself a schema mapping in which the source schema is the same as the target schema. An important property of $\to$ that we shall often use is that $\to \circ \to = \to$.

We say that J is a *solution for* I with respect to $\mathcal{M}$ if $(I, J) \in \mathcal{M}$. We use $\mathrm{Sol}_{\mathcal{M}}(I)$ to denote the set of all solutions of I with respect to $\mathcal{M}$. We say that J is a *universal solution for* I with respect to $\mathcal{M}$ if $J \in \mathrm{Sol}_{\mathcal{M}}(I)$ and for every $J' \in \mathrm{Sol}_{\mathcal{M}}(I)$, we have $J \to J'$.

Dependencies and schema mappings. A *source-to-target tuple-generating dependency*, in short, an *s-t tgd*, is a first-order formula of the form $\forall \mathbf{x}(\varphi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}))$, where $\varphi(\mathbf{x})$ is a conjunction of atomic formulas over $\mathbf{S}$, $\psi(\mathbf{x}, \mathbf{y})$ is a conjunction of atomic formulas over $\mathbf{T}$, and every variable in $\mathbf{x}$ occurs in an atomic formula in $\varphi(\mathbf{x})$. A *full* s-t tgd is a tgd with no existential quantifiers $\exists \mathbf{y}$. In this article, we will focus on schema mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ with Σ a finite set of s-t tgds. Our goal is to study *extended inverses* and *maximum extended recoveries* of such schema mappings. In particular, we will also investigate the language needed to express maximum extended recoveries. In the full case, this will require bringing into the picture a richer class of dependencies (disjunctive tgds with inequalities) that was first considered in the study of quasi-inverses of schema mappings [Fagin et al. 2008] and then in the study of maximum recoveries [Arenas et al. 2009]. In the general (not necessarily full) case, we make use of existential second-order logic (as was done in Arenas et al. [2009] for maximum recoveries).

Let *Constant* be a relation symbol that is different from all relation symbols in $\mathbf{S}$ and $\mathbf{T}$. A *disjunctive tgd with constants and inequalities from* $\mathbf{T}$ *to* $\mathbf{S}$ is a first-order formula:

$$\forall \mathbf{x}(\varphi(\mathbf{x}) \to \bigvee_{i=1}^{n} \exists \mathbf{y}_i \psi_i(\mathbf{x}, \mathbf{y}_i)),$$

where:

—The formula $\varphi(\mathbf{x})$ is a conjunction of:
  (1) atoms over $\mathbf{T}$, such that every variable in $\mathbf{x}$ occurs in at least one of them;
  (2) formulas of the form $Constant(x)$ with x a variable in $\mathbf{x}$;
  (3) inequalities $x \neq x'$ with x and x' variables in $\mathbf{x}$.

—Each formula $\psi_i(\mathbf{x}, \mathbf{y}_i)$ is a conjunction of atoms over $\mathbf{S}$.

Naturally, a formula $Constant(x)$ evaluates to true if and only if x is instantiated by a value in $\underline{\mathsf{Const}}$.

In this article, we also make use of *disjunctive tgds with inequalities*, which are obtained by not allowing condition (2) in the preceding definition. Moreover, if neither condition (2) nor (3) is allowed, then we refer to these formulas simply as *disjunctive tgds*. We shall also make use of *tgds with constants*, which are obtained by disallowing both disjunction and condition (3).

Composing and inverting schema mappings. Next, we recall the concept of the *composition* of two schema mappings, introduced in Fagin et al. [2005b] and Melnik [2004], and the concept of an *inverse* of a schema mapping, introduced in Fagin [2007].

If $\mathcal{M}_{12} = (\mathbf{S}_1, \mathbf{S}_2, \Sigma_{12})$ and $\mathcal{M}_{23} = (\mathbf{S}_2, \mathbf{S}_3, \Sigma_{23})$ are two schema mappings, their *composition* $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is a schema mapping $(\mathbf{S}_1, \mathbf{S}_3, \Sigma_{13})$ such that for every $\mathbf{S}_1$-instance I and every $\mathbf{S}_3$-instance K, we have that $(I, K) \models \Sigma_{13}$ if and only if there is a $\mathbf{S}_2$-instance J such that $(I, J) \models \Sigma_{12}$ and $(J, K) \models \Sigma_{23}$. When the schemas involved are understood from the context, we will often write $\Sigma_{12} \circ \Sigma_{23}$ to denote the composition $\mathcal{M}_{12} \circ \mathcal{M}_{23}$.

The study of inverses of schema mappings in Fagin [2007] assumes that source instances are ground. Let $\hat{\mathbf{S}}$ be a replica of the source schema $\mathbf{S}$. That is, for every relation symbol R of $\mathbf{S}$, the schema $\hat{\mathbf{S}}$ contains a relation symbol $\hat{R}$ that is not in $\mathbf{S}$ and has the same arity as R. Thus, every source instance I has a replica instance $\hat{I}$ over $\hat{\mathbf{S}}$.

The *identity schema mapping* is $\text{Id} = (\mathbf{S}, \hat{\mathbf{S}}, \Sigma_{\text{Id}})$, where Σ_{Id} consists of the dependencies $R(\mathbf{x}) \rightarrow \hat{R}(\mathbf{x})$ as R ranges over the relation symbols in $\mathbf{S}$. Thus, from the semantic point of view, Id is the set of all pairs (I_1, I_2) such that I_1 is a ground $\mathbf{S}$-instance, I_2 is a ground $\hat{\mathbf{S}}$-instance, and $\hat{I}_1 \subseteq I_2$. We note that the identity mapping is sometimes called the *copy mapping*.

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping. We say that a schema mapping $\mathcal{M}' = (\mathbf{T}, \hat{\mathbf{S}}, \Sigma')$ is an *inverse* of $\mathcal{M}$ if $\mathcal{M} \circ \mathcal{M}' = \text{Id}$ (as sets of pairs of instances). This means that, for every pair (I_1, I_2) of a ground $\mathbf{S}$-instance I_1 and a ground $\hat{\mathbf{S}}$-instance I_2, we have that $\hat{I}_1 \subseteq I_2$ if and only if there is a target instance J such that $(I_1, J) \models \Sigma$ and $(J, I_2) \models \Sigma'$.

From now on and for notational simplicity, we will write $\mathbf{S}$ to also denote its replica $\hat{\mathbf{S}}$; it will be clear from the context if we refer to $\mathbf{S}$ or to its replica. Moreover, we will use the same symbol to denote both a ground $\mathbf{S}$-instance I and its replica $\hat{\mathbf{S}}$-instance $\hat{I}$.

3. EXTENDED INVERSES

In this section, we introduce and study the notion of an extended inverse of a schema mapping. For this, we first need the notions of an extended solution and of the extended identity mapping.

Definition 3.1. Let $\mathcal{M}$ be a schema mapping. We say that J is an *extended solution* of I with respect to $\mathcal{M}$ if there exist instances I' and J' such that $I \rightarrow I'$, $(I', J') \in \mathcal{M}$, and $J' \rightarrow J$. This is the same as saying that $(I, J) \in \rightarrow \circ \mathcal{M} \circ \rightarrow$. We use $\text{eSol}_{\mathcal{M}}(I)$ to denote the set of all extended solutions of I with respect to $\mathcal{M}$.

Example 3.2. Recall the earlier Example 1.1. The target instance $U = \{Q(a, b), R(b, c)\}$ is not a solution for the source instance $V = \{P(a, b, Z), P(X, b, c)\}$ with respect to schema mapping $\mathcal{M}$ because every solution for V with respect to $\mathcal{M}$ must contain $R(b, Z)$ and $Q(X, b)$. However, U is an extended solution for V with respect to $\mathcal{M}$. To see this, consider the target instance:

$$U' = \{Q(a, b), Q(X, b), R(b, c), R(b, Z)\},$$

which is a solution for V with respect to $\mathcal{M}$. Furthermore, there is a homomorphism from U' to U (where X is mapped to a, and Z is mapped to c). Thus, $(V, U') \in \mathcal{M}$, and $U' \to U$. Hence, U is an extended solution for V with respect to $\mathcal{M}$.

Another way to see that U is an extended solution for V with respect to $\mathcal{M}$ is to observe that if I is as in Example 1.1, then $V \to I$, and U itself is a solution for I with respect to $\mathcal{M}$.

As this example illustrates, the main idea behind extended solutions is that, since nulls represent unknown information, it is legal to homomorphically map them to other values either before or after taking the standard notion of solution.

The following proposition shows that extended solutions coincide with solutions in an important special case.

PROPOSITION 3.3. *If I is a ground source instance and $\mathcal{M}$ is a schema mapping specified by s-t tgds, then* $\mathrm{eSol}_{\mathcal{M}}(I) = \mathrm{Sol}_{\mathcal{M}}(I)$.

PROOF. By the definitions of $\mathrm{Sol}_{\mathcal{M}}(I)$ and $\mathrm{eSol}_{\mathcal{M}}(I)$, it is immediate that $\mathrm{Sol}_{\mathcal{M}}(I) \subseteq \mathrm{eSol}_{\mathcal{M}}(I)$. Next, we will show that the reverse inclusion also holds. Assume that $J \in \mathrm{eSol}_{\mathcal{M}}(I)$. Hence, there exists I' and J' such that $I \to I'$, $(I', J') \in \mathcal{M}$, and $J' \to J$. Since I is a ground instance, we know that $I \subseteq I'$. Therefore, since $(I', J') \in \mathcal{M}$, it follows that $(I, J') \in \mathcal{M}$ (this property of tgds is sometimes referred to as being closed down on the left). Since $J' \to J$ and since s-t tgds are preserved under target homomorphisms, it follows that J is a solution for I with respect to $\mathcal{M}$. □

Based on extended solutions, we define extended universal solutions by mimicking the definition of universal solutions in Fagin et al. [2005a].

Definition 3.4. Let $\mathcal{M}$ be a schema mapping. We say that J is an *extended universal solution* for I with respect to $\mathcal{M}$ if $J \in \mathrm{eSol}_{\mathcal{M}}(I)$ and, for every $J' \in \mathrm{eSol}_{\mathcal{M}}(I)$, we have that $J \to J'$.

We now define the notion of a homomorphic extension of a schema mapping. This plays a central role in what follows.

Definition 3.5. Let $\mathcal{M}$ be a schema mapping. The *homomorphic extension of* $\mathcal{M}$, denoted by $e(\mathcal{M})$, is the schema mapping $\to \circ \mathcal{M} \circ \to$.

Note that for every source instance I, the extended solutions for I with respect to $\mathcal{M}$ are exactly the standard solutions of I with respect to $e(\mathcal{M})$.

In the same spirit as the extended notion of solution, we now consider an extended notion of the identity schema mapping. This is obtained by applying the homomorphic extension operator e on the standard identity schema mapping Id.

Definition 3.6. The *extended identity schema mapping* is the schema mapping $e(\mathrm{Id})$.

Note that, by definition, $e(\mathrm{Id}) = \to \circ \mathrm{Id} \circ \to$. It is easy to see that $\to \circ \mathrm{Id} \circ \to$ is the same as $\to$, and therefore, $e(\mathrm{Id}) = \to$. Thus, the key difference from the standard notion of the identity schema mapping is that $e(\mathrm{Id})$ considers pairs (I_1, I_2) of instances such that I_1 is not necessarily a subset of I_2, but instead, I_1 can be homomorphically mapped into I_2. Intuitively, I_1 is a subset of I_2 up to homomorphic mapping of nulls. Note that when I_1 and I_2 are ground, we have that $I_1 \rightarrow I_2$ if and only if $I_1 \subseteq I_2$. Thus, for ground instances, $e(\mathrm{Id})$ coincides with Id.

We are now ready to give the definition of an extended inverse of a schema mapping.

Definition 3.7. Let $\mathcal{M}$ be a schema mapping.

—A schema mapping $\mathcal{M}'$ is an *extended inverse* of $\mathcal{M}$ if

$$e(\mathcal{M}) \circ e(\mathcal{M}') = e(\mathrm{Id}).$$

Since $\rightarrow \circ \rightarrow = \rightarrow$, this equation is the same as

$$\rightarrow \circ \mathcal{M} \circ \rightarrow \circ \mathcal{M}' \circ \rightarrow = \rightarrow.$$

—$\mathcal{M}$ is *extended-invertible* if it has an extended inverse. □

We next introduce the notion of a *capturing function*, which will be used to characterize extended invertibility.

Definition 3.8. Let $\mathcal{M}$ be a schema mapping.

—We say that a target instance J *captures* a source instance I (for $\mathcal{M}$) if the following two conditions hold: (a) $J \in \mathrm{eSol}_{\mathcal{M}}(I)$; and (b) if K is a source instance such that $J \in \mathrm{eSol}_{\mathcal{M}}(K)$, then $K \rightarrow I$.
—A *capturing function* for $\mathcal{M}$ is a (total) function F from source instances to target instances such that for every source instance I, we have that $F(I)$ captures I.

The definitions imply that if J captures both I_1 and I_2, then I_1 and I_2 are homomorphically equivalent. Thus, the source instance that is captured by J is unique up to homomorphic equivalence. In general, for a given I, there may not exist a J that captures I. If such J exists for every I, then a capturing function for $\mathcal{M}$ exists.

We note that the notion of a target instance capturing a source instance is an extended version of the notion of a strong witness solution in Arenas et al. [2009]. We now give a lemma that will be used in the following theorem, which shows that extended invertibility is equivalent to the existence of a capturing function.

LEMMA 3.9. *Assume that the schema mapping $\mathcal{M}'$ is an extended inverse of the schema mapping $\mathcal{M}$. If $(I, J) \in e(\mathcal{M})$ and $(J, I) \in e(\mathcal{M}')$, then J captures I.*

PROOF. Assume that K is a source instance such that $J \in \mathrm{eSol}_{\mathcal{M}}(K)$; we must show that $K \rightarrow I$. Since $J \in \mathrm{eSol}_{\mathcal{M}}(K)$, it follows that $(K, J) \in e(\mathcal{M})$. Since also $(J, I) \in e(\mathcal{M}')$, we have that $(K, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. Therefore, since $e(\mathcal{M}) \circ e(\mathcal{M}') = e(\mathrm{Id})$, it follows that $(K, I) \in e(\mathrm{Id})$, and so $K \rightarrow I$, as desired. □

THEOREM 3.10. *Let $\mathcal{M}$ be a schema mapping. The following statements are equivalent.*

(1) *$\mathcal{M}$ is extended-invertible.*
(2) *There exists a capturing function for $\mathcal{M}$.*

Furthermore, if $\mathcal{M}$ is extended-invertible and F is a capturing function for $\mathcal{M}$, then $\mathcal{M}' = \{(J, I) \mid J = F(I)\}$ is an extended inverse of $\mathcal{M}$.

PROOF. Assume first that $\mathcal{M}$ is extended-invertible; we will show that there exists a capturing function for $\mathcal{M}$. Let $\mathcal{M}'$ be an extended inverse of $\mathcal{M}$.

Let I be an arbitrary source instance. We shall show that there is a target instance J that captures I, and we can then define the value $F(I)$ of the capturing function F to be J. Since $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$, we know that $e(\mathcal{M}) \circ e(\mathcal{M}') = e(\mathrm{Id})$. Since

$(I, I) \in e(\text{Id})$, it follows that $(I, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. So there is J such that $(I, J) \in e(\mathcal{M})$ and $(J, I) \in e(\mathcal{M}')$. By Lemma 3.9, it follows that J captures I.

Assume now that there exists a capturing function F for $\mathcal{M}$, Let $\mathcal{M}' = \{(J, I) \mid J = F(I)\}$. The proof is concluded if we show that $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$, that is, that $e(\mathcal{M}) \circ e(\mathcal{M}') = e(\text{Id})$.

Assume first that $(I_1, I_2) \in e(\text{Id})$, that is, $I_1 \rightarrow I_2$; we must show that $(I_1, I_2) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. Let $J = F(I_2)$. By definition of $\mathcal{M}'$, we have that $(J, I_2) \in \mathcal{M}'$. Since J captures I_2, we have in particular that $(I_2, J) \in e(\mathcal{M})$, that is, $(I_2, J) \in \rightarrow \circ \mathcal{M} \circ \rightarrow$. Since also $I_1 \rightarrow I_2$, we have $(I_1, J) \in \rightarrow \circ \rightarrow \circ \mathcal{M} \circ \rightarrow = \rightarrow \circ \mathcal{M} \circ \rightarrow = e(\mathcal{M})$ (with the first equality holding since $\rightarrow \circ \rightarrow = \rightarrow$). Since $(I_1, J) \in e(M)$ and $(J, I_2) \in \mathcal{M}' \subseteq e(M')$, we have $(I_1, I_2) \in e(M) \circ e(\mathcal{M}')$, as desired.

Assume now that $(I_1, I_2) \in e(M) \circ e(\mathcal{M}')$; we must show that $(I_1, I_2) \in e(\text{Id})$, that is, that $I_1 \rightarrow I_2$. Since $(I_1, I_2) \in e(M) \circ e(\mathcal{M}') = \rightarrow \circ \mathcal{M} \circ \rightarrow \circ \mathcal{M}' \rightarrow = e(\mathcal{M}) \circ (\mathcal{M}' \circ \rightarrow)$, there is J such that $(I_1, J) \in e(M)$ and $(J, I_2) \in \mathcal{M}' \circ \rightarrow$. Since $(J, I_2) \in \mathcal{M}' \circ \rightarrow$, there is I_3 such that $(J, I_3) \in \mathcal{M}'$ and $I_3 \rightarrow I_2$. Since $(J, I_3) \in \mathcal{M}'$, it follows from the definition of $\mathcal{M}'$ that J captures I_3. Since J captures I_3, and since $(I_1, J) \in e(\mathcal{M})$, we have that $I_1 \rightarrow I_3$. But we also have that $I_3 \rightarrow I_2$, so by composing homomorphisms we see that $I_1 \rightarrow I_2$, as desired. □

3.1. Extended Inverses of Mappings Specified by s-t tgds

We now address the important case in which the schema mapping $\mathcal{M}$ is specified by a finite set of s-t tgds. First we relate extended-invertibility of such schema mappings to the existence of a special capturing function given by the chase, and also to a homomorphism-based property, which we define shortly. We consider here the standard chase with tgds as introduced in Beeri and Vardi [1984] (see also Abiteboul et al. [1995]) but applied to data exchange [Fagin et al. 2005a]. In particular, given a source instance I, we write $\mathit{chase}_{\mathcal{M}}(I)$ to denote a target instance J such that (I, J) is the result of chasing $(I, \emptyset)$ with the dependencies in $\mathcal{M}$. Note here that $(I, \emptyset)$ and (I, J) are instances over the combined source and target schema.

Since the chase is rather well-known, we do not include its basic definition here. Note that there are different variants of the chase, and each variant may produce nonisomorphic results depending on the chase sequence followed. However, all these different outputs are homomorphically equivalent to each other, and the choice of a particular output has no effect on the concepts and results. Note also that the reader can find the more general definition of the *disjunctive chase* in Section 6, where it is used for reverse data exchange.

The following proposition is analogous to (and follows easily from) the result in Fagin et al. [2005a] that $\mathit{chase}_{\mathcal{M}}(I)$ is a universal solution for I.

PROPOSITION 3.11. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. If J is a universal solution for I with respect to $\mathcal{M}$, then J is an extended universal solution for I with respect to $\mathcal{M}$. In particular, $\mathit{chase}_{\mathcal{M}}(I)$ is an extended universal solution for I.*

PROOF. Let J be a universal solution for I, and let J' be an extended solution for I. We must show that $J \rightarrow J'$. Since J' is an extended solution for I, there are I_1 and J_1 such that $I \rightarrow I_1$, $(I_1, J_1) \in \mathcal{M}$, and $J_1 \rightarrow J'$. Since $I \rightarrow I_1$, it follows that $\mathit{chase}_{\mathcal{M}}(I) \rightarrow \mathit{chase}_{\mathcal{M}}(I_1)$. Since J is a universal solution for I, and since $\mathit{chase}_{\mathcal{M}}(I)$ is a solution for I, it follows that $J \rightarrow \mathit{chase}_{\mathcal{M}}(I)$. Since $\mathit{chase}_{\mathcal{M}}(I_1)$ is a universal solution for I_1, and since $(I_1, J_1) \in \mathcal{M}$, it follows that $\mathit{chase}_{\mathcal{M}}(I_1) \rightarrow J_1$. Moreover, $J_1 \rightarrow J'$, so by composing these four homomorphisms, we have $J \rightarrow J'$, as desired. □

We now show that when $\mathcal{M}$ is extended invertible and J is an extended universal solution for a source instance I according to $\mathcal{M}$, then J captures I. However, the converse may not be true, even when $\mathcal{M}$ is specified by s-t tgds.

PROPOSITION 3.12. *Let $\mathcal{M}$ be a schema mapping that is extended invertible. If J is an extended universal solution for I, then J captures I.*

PROOF. First, we have that J is an extended solution for I. Next, let K be a source instance such that $J \in \mathrm{eSol}_{\mathcal{M}}(K)$; we must show that $K \rightarrow I$. Since $\mathcal{M}$ is extended invertible, it follows from Theorem 3.10 that there is J^* that captures I. Since J^* captures I, it follows in particular that J^* is an extended solution for I, and so $J \rightarrow J^*$ by universality of J. Since $J \in \mathrm{eSol}_{\mathcal{M}}(K)$, we have that $(K, J) \in \rightarrow \circ\mathcal{M}\circ \rightarrow$. Combining this with $J \rightarrow J^*$, we have that $(K, J^*) \in \rightarrow \circ\mathcal{M}\circ \rightarrow \circ \rightarrow = \rightarrow \circ\mathcal{M}\circ \rightarrow$, with the equality holding, since $\rightarrow \circ \rightarrow = \rightarrow$. Therefore, $J^* \in \mathrm{eSol}_{\mathcal{M}}(K)$. Since also J^* captures I, it follows that $K \rightarrow I$, as desired. □

Note that we do not assume in Proposition 3.12 that $\mathcal{M}$ is specified by s-t tgds. We now show that the converse to Proposition 3.12 fails, even if $\mathcal{M}$ is specified by s-t tgds.

PROPOSITION 3.13. *There exists an extended-invertible schema mapping specified by an s-t tgd, and there exist a source instance I and a target instance J, such that J captures I, but J is not an extended universal solution for I.*

PROOF. Consider the schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where the source schema $\mathbf{S}$ consists of one unary relation symbol P, the target schema $\mathbf{T}$ consists of one binary relation symbol Q, and Σ consists of the following s-t tgd:

$$P(x) \rightarrow \exists z(Q(x, z) \wedge Q(z, x)).$$

To show that $\mathcal{M}$ is extended invertible, consider the schema mapping $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$, where Σ' consists of the following s-t tgd:

$$Q(x, z) \wedge Q(z, x) \rightarrow P(x).$$

In Section 3.2, we shall define and study the notion of a *chase-inverse*. There we shall show (Example 3.24) that $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$, by showing that $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$, and showing that this implies that $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$.

Let I be the source instance $\{P(0)\}$, and let J be the target instance $\{Q(0, 0)\}$. We now show that J captures I, but J is not an extended universal solution for I.

It is easy to see that $(I, J) \in \mathcal{M}$ and $(J, I) \in \mathcal{M}'$. So by Lemma 3.9, it follows that J captures I. Let J' be $\{Q(0, n), Q(n, 0)\}$, where n is a null value. Clearly, J' is a solution for I, but there is no homomorphism from J into J'. So J is not an extended universal solution for I. □

Definition 3.14. Assume that $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds. We say that $\mathcal{M}$ has the *homomorphism property* if for all source instances I_1 and I_2, the following holds: if $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$, then $I_1 \rightarrow I_2$.

Note that when $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds, then the converse of the homomorphism property always holds. That is, if $I_1 \rightarrow I_2$, then it follows that $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$.

THEOREM 3.15. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. The following statements are equivalent.*

(1) *$\mathcal{M}$ is extended-invertible.*
(2) *There exists a capturing function for $\mathcal{M}$.*

(3) *The function F with $F(I) = chase_{\mathcal{M}}(I)$ is a capturing function for $\mathcal{M}$.*
(4) *$\mathcal{M}$ has the homomorphism property.*

Moreover, if $\mathcal{M}$ is extended-invertible, then the schema mapping $\mathcal{M}^* = \{(J, I) \mid J = chase_{\mathcal{M}}(I)\}$ is an extended inverse of $\mathcal{M}$.

PROOF. The equivalence of (1) and (2) follows from Theorem 3.10. It is immediate that (3) ⇒ (2). We now show that (1) ⇒ (3), and so (1), (2), and (3) are equivalent. We know from Fagin et al. [2005a] that $chase_{\mathcal{M}}(I)$ is a universal solution for I. So by Proposition 3.11, it follows that $chase_{\mathcal{M}}(I)$ is an extended universal solution for I. Therefore, by Proposition 3.12, we have that $chase_{\mathcal{M}}(I)$ captures I. So (3) follows.

We now show that (3) ⇒ (4). Assume that (3) holds. Assume that $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$; we must show that $I_1 \rightarrow I_2$. Now $(I_1, chase_{\mathcal{M}}(I_1)) \in \mathcal{M}$, and therefore $(I_1, chase_{\mathcal{M}}(I_2)) \in \mathcal{M} \circ \rightarrow$. Hence, $chase_{\mathcal{M}}(I_2) \in \text{eSol}_{\mathcal{M}}(I_1)$. By (3), we have that $chase_{\mathcal{M}}(I_2)$ captures I_2. Therefore, since $chase_{\mathcal{M}}(I_2) \in \text{eSol}_{\mathcal{M}}(I_1)$, we have $I_1 \rightarrow I_2$, as desired.

We conclude the proof by showing that (4) ⇒ (3). Assume that the homomorphism property holds; we must show that $chase_{\mathcal{M}}(I)$ captures I. First, $chase_{\mathcal{M}}(I)$ is a solution for I, and hence an extended solution for I. Assume now that $chase_{\mathcal{M}}(I) \in \text{eSol}_{\mathcal{M}}(K)$; the proof is complete if we show that $K \rightarrow I$. Now $chase_{\mathcal{M}}(K)$ is a universal solution for K; hence, $chase_{\mathcal{M}}(K)$ is an extended universal solution for K by Proposition 3.11. Therefore, since $chase_{\mathcal{M}}(I) \in \text{eSol}_{\mathcal{M}}(K)$, we have that $chase_{\mathcal{M}}(K) \rightarrow chase_{\mathcal{M}}(I)$. So by the homomorphism property, we have $K \rightarrow I$, as desired. □

Theorems 3.10 and 3.15 provide useful tools for verifying whether a schema mapping is extended invertible or not. Next, we give an example that illustrates how Theorem 3.15 can be applied to show that a schema mapping specified by s-t tgds is not extended invertible.

Example 3.16. Consider the "union" schema mapping $\mathcal{M}$ specified by the s-t tgds $P(x) \rightarrow R(x)$ and $Q(x) \rightarrow R(x)$. We now prove that $\mathcal{M}$ is not extended-invertible by showing that $\mathcal{M}$ does not have the homomorphism property. Let $I_1 = \{P(0)\}$ and $I_2 = \{Q(0)\}$. Clearly, $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$; however, $I_1 \not\rightarrow I_2$.

We next relate the notions of extended invertibility and extended inverses to the notions of invertibility and inverses.

THEOREM 3.17. *The following hold.*

(1) *If $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds and $\mathcal{M}$ is extended invertible, then $\mathcal{M}$ is invertible.*
(2) *There is a schema mapping $\mathcal{M}$ specified by a finite set of s-t tgds that is invertible but not extended-invertible.*
(3) *There is a schema mapping $\mathcal{M}$ specified by a finite set of s-t tgds that is extended-invertible and such that:*
    (a) *$\mathcal{M}$ has an extended inverse $\mathcal{M}'$ that is not an inverse of $\mathcal{M}$.*
    (b) *$\mathcal{M}$ has an inverse $\mathcal{M}''$ that is not an extended inverse of $\mathcal{M}$.*

PROOF. Part (1) follows from the fact that the homomorphism property, which by Theorem 3.15 is equivalent to extended invertibility, implies the subset property (defined next), which by Corollary 3.23 of Fagin et al. [2008] is equivalent to invertibility.

A schema mapping $\mathcal{M}$ has the *subset property* (which is referred to in Fagin et al. [2008] as the $(=,=)$-*subset property*) if for all ground instances I_1 and I_2 such that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$, necessarily $I_1 \subseteq I_2$. It is shown in Fagin et al. [2008] that a schema mapping specified by a finite set of s-t tgds has the subset property if and only if it is invertible.

Assume that $\mathcal{M}$ is an extended-invertible schema mapping specified by a finite set of s-t tgds. Therefore, by Theorem 3.15, it follows that $\mathcal{M}$ has the homomorphism property. We now show that $\mathcal{M}$ has the subset property, which will complete the proof of part (1). Assume that I_1 and I_2 are ground instances such that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$; we must show that $I_1 \subseteq I_2$. Since $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$; we have in particular that $chase_{\mathcal{M}}(I_2) \in \text{Sol}_{\mathcal{M}}(I_1)$. Since $chase_{\mathcal{M}}(I_1)$ is a universal solution for I_1, it follows that $chase_{\mathcal{M}}(I_1) \to chase_{\mathcal{M}}(I_2)$. So by the homomorphism property, it follows that $I_1 \to I_2$. Since I_1 is a ground instance, this implies that $I_1 \subseteq I_2$, as desired.

For part (2), consider the schema mapping $\mathcal{M}$ that is specified by the following s-t tgds:

$$P(x) \to \exists y R(x, y) \qquad Q(y) \to \exists x R(x, y).$$

We now show that $\mathcal{M}$ is invertible but not extended-invertible. First, it can be easily verified that the schema mapping $\mathcal{M}'$ specified by the following s-t tgds with constants is an inverse of $\mathcal{M}$:

$$R(x, y) \wedge Constant(x) \to P(x)$$
$$R(x, y) \wedge Constant(y) \to Q(y)$$

Thus, $\mathcal{M}$ is invertible. We now show that $\mathcal{M}$ is not extended-invertible by showing it fails to satisfy the homomorphism property. Consider the source instances $I_1 = \{P(n_1)\}$ and $I_2 = \{Q(n_2)\}$, where n_1 and n_2 are nulls. Then $chase_{\mathcal{M}}(I_1)$ and $chase_{\mathcal{M}}(I_2)$ are homomorphically equivalent. In particular, $chase_{\mathcal{M}}(I_1) \to chase_{\mathcal{M}}(I_2)$. However, it is not the case that $I_1 \to I_2$.

For part (3), consider the schema mapping $\mathcal{M}$ specified by the following s-t tgd:

$$P(x, y) \to \exists z(Q(x, z) \wedge Q(z, y)).$$

Moreover, consider the following reverse schema mappings $\mathcal{M}'$ and $\mathcal{M}''$ given, respectively, by the following dependencies:

$$\begin{aligned} \mathcal{M}' :& \quad Q(x, z) \wedge Q(z, y) \to P(x, y) \\ \mathcal{M}'' :& \quad Q(x, z) \wedge Q(z, y) \wedge Constant(x) \wedge Constant(y) \\ & \qquad \to P(x, y). \end{aligned}$$

We shall show in the next subsection (Example 3.23) that $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$. At the same time, it was shown in Fagin et al. [2008] that there is no inverse of $\mathcal{M}$ that is specified by s-t tgds without the *Constant* predicate. Hence, $\mathcal{M}'$ cannot be an inverse of $\mathcal{M}$. Thus, $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$ that is not an inverse of $\mathcal{M}$. As for condition (b), it was shown in Fagin et al. [2008] that $\mathcal{M}''$ is an inverse of $\mathcal{M}$. We show in the next subsection (Example 3.25) that $\mathcal{M}''$ is not an extended inverse of $\mathcal{M}$. □

Theorem 3.17 tells us that the notion of extended invertibility is stronger than invertibility. Intuitively, it is harder for a schema mapping to be extended invertible, since there are more instances (nonground source instances) to consider. Furthermore, extended inverses and inverses do not necessarily coincide, even for schema mappings that are extended-invertible (hence, also invertible).

3.2. The Goodness of Extended Inverses

In this section, we show that extended inverses that are specified by s-t tgds have desirable properties that make them ideally suited for reverse data exchange. The data exchange problem associated with a schema mapping $\mathcal{M}$ is to materialize a "good" solution, J, from a source instance I, based on $\mathcal{M}$. The canonical procedure for data exchange [Fagin et al. 2005a] uses the chase of I with $\mathcal{M}$ to produce a canonical universal solution. The *reverse data exchange problem* is then to materialize a source instance I' from a target instance J according to a reverse schema mapping $\mathcal{M}'$ (from the target schema to the source schema). Reverse data exchange is typically performed after an initial data exchange with $\mathcal{M}$ was performed; in such a case, the goal of reverse data exchange is to recover a source instance that is as "close as possible" to the *original* source instance I.

Definition 3.18. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$ be schema mappings, where Σ and Σ' are finite sets of s-t tgds. We say that $\mathcal{M}'$ is a *chase-inverse* of $\mathcal{M}$ if I and $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$ are homomorphically equivalent, for every source instance I.

Note that a chase-inverse makes it possible to recover the original source instance up to homomorphic equivalence. Fagin and Nash [2010] showed that if $\mathcal{M}$ is an invertible schema mapping specified by s-t tgds, then $\mathcal{M}$ has a *normal* inverse $\mathcal{M}'$ such that $I = chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$, for every source instance I. A *normal* inverse is an inverse specified by s-t tgds with constants and inequalities, and with certain syntactic restrictions. In contrast, our Definition 3.18 requires only homomorphic equivalence, not equality, of I and $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$.

We now show that if there is a chase-inverse, then there is a full chase-inverse, that is a chase-inverse specified by a finite set of full s-t tgds. This is analogous to (and with a similar proof as) the result of Fagin [2007] that if a schema mapping specified by a finite set of s-t tgds has an inverse specified by a finite set of s-t tgds, then it has an inverse specified by a finite set of full s-t tgds.

We begin with some definitions from Fagin [2007]. Let γ be the tgd $\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \exists \mathbf{y}\psi(\mathbf{x}, \mathbf{y}))$. Assume that all of the variables in $\mathbf{x}$ appear in $\phi(\mathbf{x})$ (but not necessarily in $\psi(\mathbf{x}, \mathbf{y})$). Let $\psi^f(\mathbf{x})$ be the conjunction of all atoms in $\psi(\mathbf{x}, \mathbf{y})$ that do not contain any variables in $\mathbf{y}$ (the f stands for "full"). Define γ^f to be the full tgd $\forall \mathbf{x}(\phi(\mathbf{x}) \rightarrow \psi^f(\mathbf{x}))$. As an example (where we do not bother to write the universal quantifiers), if γ is $P(x_1, x_2) \rightarrow \exists y(Q(x_1, x_1) \wedge Q(x_2, y))$, then γ^f is $P(x_1, x_2) \rightarrow Q(x_1, x_1)$. If $\psi^f(\mathbf{x})$ is an empty conjunction, then γ^f is a *dummy tgd* where the conclusion is "Truth" (and so the dummy tgd itself is "Truth"). Let $\mathcal{M}_1 = (\mathbf{T}, \mathbf{S}, \Sigma_1)$ be a schema mapping, where Σ_1 is a finite set of s-t tgds (our case of interest is where $\mathcal{M}_1$ is a chase-inverse). Let Σ_1^f be the set of γ^f where $\gamma \in \Sigma_1$ and where γ^f is not a dummy tgd, and let $\mathcal{M}_1^f = (\mathbf{T}, \mathbf{S}, \Sigma_1^f)$

THEOREM 3.19. *Assume that $\mathcal{M}$ and $\mathcal{M}_1$ are schema mappings, each specified by a finite set of s-t tgds, where $\mathcal{M}_1$ is a chase-inverse of $\mathcal{M}$. Then $\mathcal{M}_1^f$ is also a chase-inverse of $\mathcal{M}$.*

PROOF. Since $\mathcal{M}_1$ is a chase-inverse of $\mathcal{M}$, we know that I and $chase_{\mathcal{M}_1}(chase_{\mathcal{M}}(I))$ are homomorphically equivalent, for every source instance I. We must show that I and $chase_{\mathcal{M}_1^f}(chase_{\mathcal{M}}(I))$ are homomorphically equivalent, for every source instance I. Since $chase_{\mathcal{M}_1^f}(chase_{\mathcal{M}}(I)) \subseteq chase_{\mathcal{M}_1}(chase_{\mathcal{M}}(I))$, and since there is a homomorphism from $chase_{\mathcal{M}_1}(chase_{\mathcal{M}}(I))$ to I, there is a homomorphism from $chase_{\mathcal{M}_1^f}(chase_{\mathcal{M}}(I))$ to I. So we need only show that there is a homomorphism from I to $chase_{\mathcal{M}_1^f}(chase_{\mathcal{M}}(I))$. It is sufficient to show that $I \subseteq chase_{\mathcal{M}_1^f}(chase_{\mathcal{M}}(I))$.

Assume first that I is ground. Since I is ground, and since there is a homomorphism from I to $chase_{\mathcal{M}_1}(chase_{\mathcal{M}}(I))$, it follows that $I \subseteq chase_{\mathcal{M}_1}(chase_{\mathcal{M}}(I))$. But this implies the desired result that $I \subseteq chase_{\mathcal{M}_1^f}(chase_{\mathcal{M}}(I))$, since all of the facts that appear in $chase_{\mathcal{M}_1}(chase_{\mathcal{M}}(I))$ but not in $chase_{\mathcal{M}_1^f}(chase_{\mathcal{M}}(I))$ contain nulls.

Assume now that I is an arbitrary source instance (not necessarily ground). Let I' be the result of replacing every null in I by a distinct new constant. By what we just showed in the ground case, we know that $I' \subseteq chase_{\mathcal{M}_1^f}(chase_{\mathcal{M}}(I'))$. But this implies the desired result that $I \subseteq chase_{\mathcal{M}_1^f}(chase_{\mathcal{M}}(I))$, since I and I' are isomorphic (under an isomorphism that ignores whether values are constants or nulls), and computing the chase (or, in this case, the chase of the chase) using s-t tgds is a mechanical procedure that performs the same on isomorphic instances. □

We have the following immediate corollary of Theorem 3.19.

COROLLARY 3.20. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds, for which a chase-inverse exists. Then $\mathcal{M}$ has a chase-inverse specified by a finite set of full s-t tgds.*

Our next theorem shows that extended inverses that are specified by s-t tgds have an equivalent characterization as chase-inverses. This characterization is a precise measure of the goodness of extended inverses for reverse data exchange. The proof of the theorem makes use of the following basic property about the chase, which appears as Lemma 3.4 in Fagin et al. [2005a].

LEMMA 3.21 (TRIANGLE LEMMA). *Let K_1 and K_2 be instances such that K_2 is obtained from K_1 via a chase step with a tgd σ. Moreover, let K be an instance such that (1) K satisfies σ, and (2) K_1 has a homomorphism into K. Then K_2 has a homomorphism into K.*

THEOREM 3.22. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping specified by a finite set of s-t tgds, and let $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$ be a schema mapping specified by a finite set of s-t tgds. The following statements are equivalent:*

(1) *$\mathcal{M}'$ is an extended inverse of $\mathcal{M}$.*
(2) *$\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$.*

PROOF. We first show that if $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$, then $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$. Assume that $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$. Let I be a source instance, let $J = chase_{\mathcal{M}}(I)$, and let $I^* = chase_{\mathcal{M}'}(J)$. We have to show that $I \to I^*$ and $I^* \to I$.

Since $(I, J) \in \mathcal{M}$ and $(J, I^*) \in \mathcal{M}'$, we have $(I, I^*) \in \mathcal{M} \circ \mathcal{M}' \subseteq e(\mathcal{M}) \circ e(\mathcal{M}') = \to$, and so $I \to I^*$, which was to be shown.

Since $(I, I) \in \to$, we have that $(I, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. This means that there are I_1, I_2, J_1, and J_2, such that $I \to I_1$, $(I_1, J_1) \in \mathcal{M}$, $J_1 \to J_2$, $(J_2, I_2) \in \mathcal{M}'$, and $I_2 \to I$. Since $(I, \emptyset) \to (I_1, J_1)$ and $(I_1, J_1) \in \mathcal{M}$, the triangle lemma for tgds applied repeatedly implies that $(I, J) \to (I_1, J_1)$. In particular, $J \to J_1$ and so $J \to J_2$ (since $J_1 \to J_2$). So, we now have that $(J, \emptyset) \to (J_2, I_2)$ and $(J_2, I_2) \in \mathcal{M}'$, which again by the triangle lemma for tgds applied repeatedly, implies that $(J, I^*) \to (J_2, I_2)$. In particular, we have that $I^* \to I_2$. Since $I_2 \to I$, we conclude that $I^* \to I$, which was to be shown.

We now show that if $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$, then $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$. Assume that $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$. To show that $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$, we must show that $e(\mathcal{M}) \circ e(\mathcal{M}') = \to$. We first show that $e(\mathcal{M}) \circ e(\mathcal{M}') \subseteq \to$. Since $\to \circ \to = \to$, it suffices to show that $\mathcal{M} \circ \to \circ \mathcal{M}' \subseteq \to$. Assume that $(I_1, I_2) \in \mathcal{M} \circ \to \circ \mathcal{M}'$. Thus, there are J_1 and J_2 such that $(I_1, J_1) \in \mathcal{M}$, $J_1 \to J_2$, and $(J_2, I_2) \in \mathcal{M}'$. Let $J = chase_{\mathcal{M}}(I_1)$, and let $K = chase_{\mathcal{M}'}(J)$. Since $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$, we have I_1 and K are homomorphically equivalent. Since J is a universal solution for I_1 and $\mathcal{M}$, we have $J \to J_1$, and therefore $J \to J_2$. This last fact, together with the fact that $(J, \emptyset)$ chases with $\mathcal{M}'$ to (J, K), and the fact that $(J_2, I_2) \in \mathcal{M}'$, implies that $(J, K) \to (J_2, I_2)$, by applying the triangle lemma repeatedly. In particular, we have $K \to I_2$. Since I_1 and K are homomorphically equivalent, we obtain $I_1 \to I_2$. Thus, $(I_1, I_2) \in \to$.

We now show that $\to \subseteq \mathcal{M} \circ \mathcal{M}' \circ \to$. This inclusion implies $\to \subseteq e(\mathcal{M}) \circ e(\mathcal{M}')$. Let $(I_1, I_2) \in \to$. Thus, $I_1 \to I_2$. Let $J = chase_{\mathcal{M}}(I_1)$, and let $K = chase_{\mathcal{M}'}(J)$. Since $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$, we have I_1 and K are homomorphically equivalent. Since $I_1 \to I_2$, it follows that $K \to I_2$. Thus, we obtained: $(I_1, J) \in \mathcal{M}$, $(J, K) \in \mathcal{M}'$, and $K \to I_2$. It follows that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}' \circ \to$. □

Theorem 3.22 easily extends to the case when $\mathcal{M}'$ is given by tgds with constants, and we also allow these as chase-inverses. Also, Theorem 3.22 gives us another tool to verify whether a schema mapping $\mathcal{M}'$ is an extended inverse of a schema mapping $\mathcal{M}$.

Example 3.23. As in the proof of part (3) of Theorem 3.17, consider the schema mapping $\mathcal{M}$ specified by the s-t tgd:

$$P(x, y) \to \exists z(Q(x, z) \wedge Q(z, y)).$$

Let $\mathcal{M}'$ be the schema mapping specified by the tgd $Q(x, z) \wedge Q(z, y) \to P(x, y)$. We can show that $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$ by showing that $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$. Indeed, let I be a source instance, let $U = chase_{\mathcal{M}}(I)$ and let $V = chase_{\mathcal{M}'}(U)$. We shall show that $I \subseteq V$ and that $V \to I$, which imply that V and I are homomorphically equivalent.

First, we observe that every fact $P(a, b)$ in I generates (via the chase with $\mathcal{M}$) two facts in U, namely $Q(a, Z_{ab})$ and $Q(Z_{ab}, b)$, where Z_{ab} is a fresh new null, distinct for every choice of a and b. (Moreover, these are the only types of facts that are generated in V.) Then V must contain every fact $P(a, b)$ (in order to satisfy $\mathcal{M}'$ for $Q(a, Z_{ab})$ and $Q(Z_{ab}, b)$). Thus, $I \subseteq V$.

We now observe that every extra fact of V (not in I) can only be of the form $P(Z_{ab}, Z_{bc})$, arising via $\mathcal{M}'$ from two facts of U of the form $Q(Z_{ab}, b)$ and $Q(b, Z_{bc})$. But then U must contain two additional facts, $Q(a, Z_{ab})$ and $Q(Z_{bc}, c)$. At the same time, the only way to have $Q(a, Z_{ab})$ and $Q(Z_{ab}, b)$ in U is to have the fact $P(a, b)$ in I. Consider now the mapping h where $h(x) = x$ for members x of I, where $h(Z_{ab}) = a$ and $h(Z_{bc}) = b$. Then h is a homomorphism from V to I. In particular, for every extra fact $P(Z_{ab}, Z_{bc})$ in V, we have that $P(h(Z_{ab}), h(Z_{bc}))$ is $P(a, b)$, which is in I.

We have already noted that the schema mapping $\mathcal{M}'$ in the above example cannot be an inverse of $\mathcal{M}$ (since, as shown in Fagin et al. [2008], such an inverse would have to make use of the *Constant* predicate, for this particular $\mathcal{M}$). Thus, this example shows in particular that the characterization via chase-inverses does not hold for inverses (since $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$ but not an inverse of $\mathcal{M}$).

Example 3.24. Let us revisit the schema mapping $\mathcal{M}$ in the proof of Proposition 3.13. This schema mapping $\mathcal{M}$ is, specified by the s-t tgd:

$$P(x) \to \exists z(Q(x, z) \wedge Q(z, x)).$$

We now show that the schema mapping $\mathcal{M}'$ specified by:

$$Q(x, z) \wedge Q(z, x) \to P(x),$$

is an extended inverse of $\mathcal{M}$, by showing that $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$.

To do this, we must show that I is homomorphically equivalent to $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$, for every source instance I. This is clearly true if I is empty. So assume that I is nonempty, and so contains some fact $P(a)$. It is easy to see that $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$ consists of I, along with facts $P(n)$ for some nulls n that do not appear in I. Let h be a function that maps each such null onto a, and is otherwise the identity. It is clear that h is a homomorphism that maps $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$ onto I. Since also $I \subseteq chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$, we have that $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$ is homomorphically equivalent to I, as desired. So $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$.

The next example uses Theorem 3.22 to prove that a schema mapping $\mathcal{M}''$ is *not* an extended inverse of a schema mapping $\mathcal{M}$.

Example 3.25. As in the proof of part (3) of Theorem 3.17 and Example 3.23, consider the schema mapping $\mathcal{M}$ specified by:

$$P(x, y) \to \exists z(Q(x, z) \wedge Q(z, y)).$$

Moreover, let $\mathcal{M}''$ be the schema mapping specified by:

$$Q(x, z) \wedge Q(z, y) \wedge Constant(x) \wedge Constant(y) \to P(x, y).$$

It was shown in Fagin et al. [2008] that $\mathcal{M}''$ is an inverse of $\mathcal{M}$. We now show that $\mathcal{M}''$ is not an extended inverse of $\mathcal{M}$ by showing that $\mathcal{M}''$ fails to be a chase-inverse of $\mathcal{M}$. (Here we allow chase-inverses to be specified by tgds with constants, as discussed after Theorem 3.22.)

To show that $\mathcal{M}''$ is not a chase-inverse of $\mathcal{M}$, consider the source instance $I = \{P(W, Z)\}$, where W and Z are nulls. Let $U = chase_{\mathcal{M}}(I)$. Then $U = \{Q(W, Y), Q(Y, Z)\}$, where Y is a null. Further, $chase_{\mathcal{M}'}(U) = \emptyset$, since there are no constants in U. Hence, $chase_{\mathcal{M}''}(chase_{\mathcal{M}}(I))$ and I are not homomorphically equivalent.

These two examples point out problems with the notion of inverse, which do not arise for the new notion of extended inverse. Example 3.23 shows a natural "inverse" (the chase-inverse) that is not captured by the notion of inverse. Conversely, Example 3.25 shows an inverse that fails to be a chase-inverse.

Most schema mappings occurring in practice do not possess extended inverses, since they do not even possess inverses. Invertibility and extended invertibility are strong notions that, intuitively cover the case of "no information loss" in a schema mapping. To address schema mappings with information loss, which is the frequent case, we will go beyond extended invertibility and study extended recoveries in Section 4.

4. EXTENDED RECOVERIES

In this section, we consider the notion of a *recovery* and a *maximum recovery*, which were introduced in Arenas et al. [2009]. One of the key results in Arenas et al. [2009] is that every schema mapping specified by s-t tgds has a maximum recovery. We show that this result fails when source instances may contain null values. We introduce the notions of an *extended recovery* and a *maximum extended recovery*, and show that every schema mapping specified by s-t tgds has a maximum extended recovery. For schema mappings specified by s-t tgds, we give a characterization of maximum extended recoveries, and use this to capture in a quantitative way the *information loss* embodied in such a schema mapping. Intuitively, the information loss of the schema mapping $\mathcal{M}$ specified by s-t tgds measures the amount by which $\mathcal{M}$ deviates from being extended invertible. In particular, the schema mapping has no information loss if and only if it is extended invertible.

Definition 4.1. ([Arenas et al. 2009]). Let $\mathcal{M}$ be a schema mapping from a source schema $\mathbf{S}$ to a target schema $\mathbf{T}$. A schema mapping $\mathcal{M}'$ from $\mathbf{T}$ to $\mathbf{S}$ is a *recovery* of $\mathcal{M}$ if for every source instance I, the pair (I, I) is in $\mathcal{M} \circ \mathcal{M}'$. The schema mapping $\mathcal{M}'$ is a *maximum recovery* of $\mathcal{M}$ if (1) $\mathcal{M}'$ is a recovery of $\mathcal{M}$, and (2) for every recovery $\mathcal{M}''$ of $\mathcal{M}$, we have that $\mathcal{M} \circ \mathcal{M}' \subseteq \mathcal{M} \circ \mathcal{M}''$.

Thus, a maximum recovery is a recovery that is optimal among all recoveries in the sense that the composition of $\mathcal{M}$ with $\mathcal{M}'$ is the smallest among the compositions of $\mathcal{M}$ with every other recovery. Similar to the framework in Fagin [2007] and Fagin et al. [2008], the study of recoveries in Arenas et al. [2009] is carried out in the context in which source instances are ground. While some of the results in Arenas et al. [2009] continue to hold even when source instances are not restricted to be ground, certain other results do not. In particular, one of the most important results in Arenas et al.

[2009] is that every schema mapping specified by s-t tgds has a maximum recovery. However, we show in the next proposition that there is a schema mapping specified by s-t tgds with no maximum recovery when source instances can be nonground.

PROPOSITION 4.2. *Let* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping where* $\Sigma = \{P(x, y) \rightarrow \exists z(Q(x, z) \wedge Q(z, y))\}$. *Then* $\mathcal{M}$ *has no maximum recovery when source instances can be nonground.*

PROOF. Our proof makes use of the notion of a *witness solution* from Arenas et al. [2009]. An instance J over $\mathbf{T}$ is a *witness* for a ground instance I over $\mathbf{S}$ under the schema mapping $\mathcal{M}$ if for every ground source instance I' over $\mathbf{S}$, if $J \in \mathrm{Sol}_{\mathcal{M}}(I')$, then $\mathrm{Sol}_{\mathcal{M}}(I) \subseteq \mathrm{Sol}_{\mathcal{M}}(I')$. In addition, if $J \in \mathrm{Sol}_{\mathcal{M}}(I)$, then J is called a *witness solution* for I under $\mathcal{M}$.

We will show that $\mathcal{M}$ has no witness solution for a specific instance I, when source instances may be nonground. Using a straightforward generalization of Theorem 3.5 of Arenas et al. [2009] to nonground instances, we will then infer that $\mathcal{M}$ has no maximum recovery. This generalization states that a schema mapping $\mathcal{M}$ given by s-t tgds has a maximum recovery if and only if for every ground or nonground source instance I, there exists a witness solution for I under $\mathcal{M}$.

Let $I = \{P(0, 1), P(1, 0)\}$, and let J_I denote $chase_{\mathcal{M}}(I)$, which is $\{Q(0, U), Q(U, 1), Q(1, V), Q(V, 0)\}$, where U and V are nulls created by the chase. It is easy to see that any witness solution J for I under $\mathcal{M}$ must contain the tuples $Q(0, X)$, $Q(X, 1)$, $Q(1, Y)$, $Q(Y, 0)$, for some values X and Y, which may be nulls or constants. We consider four cases that cover all possibilities of values for X and Y and show that in each of the cases, J cannot be a witness solution.

Case 1. $X = Y$. In this case, J contains the following four tuples:

$$Q(0, X), Q(X, 1), Q(1, X), Q(X, 0).$$

Let $I' = \{P(0, 0), P(1, 1)\}$. Obviously, $J \in \mathrm{Sol}_{\mathcal{M}}(I')$ but $\mathrm{Sol}_{\mathcal{M}}(I) \not\subseteq \mathrm{Sol}_{\mathcal{M}}(I')$, since $J_I \in \mathrm{Sol}_{\mathcal{M}}(I)$ but $J_I \notin \mathrm{Sol}_{\mathcal{M}}(I')$.

Case 2. $X \neq Y$ and at least one of X or Y is not 0 or 1. Let $I' = \{P(X, Y), P(Y, X)\}$ and let $J' = \{Q(0, a), Q(a, 1),$
$Q(1, b), Q(b, 0)\}$, where a and b are values different from X and Y. Clearly, $J \in \mathrm{Sol}_{\mathcal{M}}(I')$ but $\mathrm{Sol}_{\mathcal{M}}(I) \not\subseteq \mathrm{Sol}_{\mathcal{M}}(I')$, since $J' \in \mathrm{Sol}_{\mathcal{M}}(I)$ but $J' \notin \mathrm{Sol}_{\mathcal{M}}(I')$.

Case 3. $X \neq Y$ and $X = 0$ and $Y = 1$. In this case, J contains the following four tuples:

$$Q(0, 0), Q(0, 1), Q(1, 1), Q(1, 0).$$

Let $I' = \{P(0, 0), P(1, 1)\}$. Obviously, $J \in \mathrm{Sol}_{\mathcal{M}}(I')$ but $\mathrm{Sol}_{\mathcal{M}}(I) \not\subseteq \mathrm{Sol}_{\mathcal{M}}(I')$, since $J_I \in \mathrm{Sol}_{\mathcal{M}}(I)$ but $J_I \notin \mathrm{Sol}_{\mathcal{M}}(I')$.

Case 4. $X \neq Y$ and $X = 1$ and $Y = 0$. This case is similar to Case 3.

Next, we define the notions of extended recovery and maximum extended recovery.

Definition 4.3. A schema mapping $\mathcal{M}'$ is an *extended recovery* of a schema mapping $\mathcal{M}$ if $(I, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$, for every source instance I.

We note that it is straightforward to verify that the condition $(I, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$ is equivalent to $e(\mathrm{Id}) \subseteq \mathrm{e}(\mathcal{M}) \circ \mathrm{e}(\mathcal{M}')$.

Definition 4.4. A schema mapping $\mathcal{M}'$ is a *maximum extended recovery* of a schema mapping $\mathcal{M}$ if (1) $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$, and (2) for every extended recovery $\mathcal{M}''$ of $\mathcal{M}$, we have $e(\mathcal{M}) \circ e(\mathcal{M}') \subseteq e(\mathcal{M}) \circ e(\mathcal{M}'')$.

In Arenas et al. [2009a] (which appeared after Fagin et al. [2009]), it was pointed out that M' is a maximum extended recovery of M if and only if $e(\mathcal{M}')$ is a maximum recovery of $e(\mathcal{M})$, where maximum recovery is the same notion as given in Arenas et al. [2009] with the difference that source instances may contain nulls.

Clearly, if $\mathcal{M}_1$ and $\mathcal{M}_2$ are two maximum extended recoveries of a schema mapping $\mathcal{M}$, then $e(\mathcal{M}) \circ e(\mathcal{M}_1) = e(\mathcal{M}) \circ e(\mathcal{M}_2)$. Thus, the quantity $e(\mathcal{M}) \circ e(\mathcal{M}')$, where $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$, is a constant $\mathcal{C}_{\mathcal{M}}$ that depends only on $\mathcal{M}$. In particular, it is independent of the choice of $\mathcal{M}'$. Furthermore, by Definition 4.3, $\mathcal{C}_{\mathcal{M}}$ is the smallest superset of $e(\mathrm{Id})$ among all sets $e(\mathcal{M}) \circ e(\mathcal{M}'')$, as $\mathcal{M}''$ ranges over the extended recoveries of $\mathcal{M}$. In a precise sense, $\mathcal{C}_{\mathcal{M}}$ is the closest we can get to extended identity schema mapping $e(\mathrm{Id})$ via extended recoveries. This discussion suggests the following definition.

Definition 4.5. Let $\mathcal{M}$ be a schema mapping that admits a maximum extended recovery $\mathcal{M}'$. Then the *information loss* of $\mathcal{M}$ is defined as the set difference $(e(\mathcal{M}) \circ e(\mathcal{M}')) \setminus \rightarrow$.

In what follows, we will show that the information loss of a schema mapping $\mathcal{M}$ specified by a finite set of s-t tgds can be characterized solely in terms of $\mathcal{M}$.

4.1. Characterization of Maximum Extended Recoveries

Definition 4.6. Let $\mathcal{M}$ be a schema mapping. We say that $I_1 \rightarrow_{\mathcal{M}} I_2$ if $\mathrm{eSol}_{\mathcal{M}}(I_2) \subseteq \mathrm{eSol}_{\mathcal{M}}(I_1)$.

The next proposition characterizes $\rightarrow_{\mathcal{M}}$ when $\mathcal{M}$ is specified by a finite set of s-t tgds.

PROPOSITION 4.7. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. For all source instances I_1, I_2, we have $I_1 \rightarrow_{\mathcal{M}} I_2$ if and only if $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$.*

PROOF. By the definition of $I_1 \rightarrow_{\mathcal{M}} I_2$, we have $\mathrm{eSol}_{\mathcal{M}}(I_2) \subseteq \mathrm{eSol}_{\mathcal{M}}(I_1)$. From Proposition 3.11, it follows that $chase_{\mathcal{M}}(I_1)$ and $chase_{\mathcal{M}}(I_2)$ are extended universal solutions for I_1 and, respectively, I_2, under $\mathcal{M}$. Since $\mathrm{eSol}_{\mathcal{M}}(I_2) \subseteq \mathrm{eSol}_{\mathcal{M}}(I_1)$, we have $chase_{\mathcal{M}}(I_2) \in \mathrm{eSol}_{\mathcal{M}}(I_1)$. By the universality of $chase_{\mathcal{M}}(I_1)$, it follows that $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$.

Now suppose $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$ and that $J \in \mathrm{eSol}_{\mathcal{M}}(I_2)$. We will show that $J \in \mathrm{eSol}_{\mathcal{M}}(I_1)$. Since $chase_{\mathcal{M}}(I_2)$ is an extended universal solution for I_2 under $\mathcal{M}$, it follows that $chase_{\mathcal{M}}(I_2) \rightarrow J$. So, $chase_{\mathcal{M}}(I_1) \rightarrow J$. Since $chase_{\mathcal{M}}(I_1)$ is an extended solution of I_1 under $\mathcal{M}$ and $chase_{\mathcal{M}}(I_1) \rightarrow J$, it follows that J is an extended solution of I_1 under $\mathcal{M}$. □

We shall show that if $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds, then $\mathcal{M}$ has a maximum extended recovery. We shall actually prove something stronger, for which we need yet another definition.

Definition 4.8. A schema mapping $\mathcal{M}'$ is a *strong maximum extended recovery* of a schema mapping $\mathcal{M}$ if (1) $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$, and (2) for every extended recovery $\mathcal{M}''$ of $\mathcal{M}$, we have $e(\mathcal{M}') \subseteq e(\mathcal{M}'')$.

Note that, by monotonicity of composition, every strong maximum extended recovery of $\mathcal{M}$ is a maximum extended recovery of $\mathcal{M}$. Thus, a strong maximum extended recovery $\mathcal{M}'$ not only minimizes $e(\mathcal{M}) \circ e(\mathcal{M}'')$ among all extended recoveries $\mathcal{M}''$, but, even more, minimizes $e(\mathcal{M}'')$ among all extended recoveries $\mathcal{M}''$.

LEMMA 4.9. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. Put*

$$\mathcal{M}^* = \{(chase_{\mathcal{M}}(I), I) \mid I \textit{ is a source instance}\}.$$

If $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$, then $\mathcal{M}^* \subseteq e(\mathcal{M}')$; equivalently, $e(\mathcal{M}^*) \subseteq e(\mathcal{M}')$.

Proof. Let I be a source instance. Since $\mathcal{M}'$ is an extended recovery, we have that $(I, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. Hence, there is a J such that $(I, J) \in e(\mathcal{M})$ and $(J, I) \in e(\mathcal{M}')$. In particular, J is an extended solution for I under $\mathcal{M}$. Since $chase_{\mathcal{M}}(I)$ is an extended universal solution for I under $\mathcal{M}$, we have that $chase_{\mathcal{M}}(I) \rightarrow J$. Thus, $(chase_{\mathcal{M}}(I), I) \in \ \rightarrow \circ\, e(\mathcal{M}') = e(\mathcal{M}')$. Finally, since $\rightarrow \circ\, e(\mathcal{M}') \circ \rightarrow\, = e(\mathcal{M}')$, we have $\mathcal{M}^* \subseteq e(\mathcal{M}')$ if and only if $e(\mathcal{M}^*) \subseteq e(\mathcal{M}')$. □

Theorem 4.10. *Every schema mapping $\mathcal{M}$ specified by a finite set of s-t tgds has a strong maximum extended recovery. Specifically, the schema mapping*

$$\mathcal{M}^* = \{(chase_{\mathcal{M}}(I), I) \mid I \textit{ is a source instance}\}$$

is a strong maximum extended recovery of $\mathcal{M}$.

Proof. $\mathcal{M}^*$ is an extended recovery of $\mathcal{M}$ because, for every source instance I, we have that $(I, I) \in \mathcal{M} \circ \mathcal{M}^* \subseteq e(\mathcal{M}) \circ e(\mathcal{M}^*)$. Now, assume that $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$. By Lemma 4.9, we have that $e(\mathcal{M}^*) \subseteq e(\mathcal{M}')$. □

We now show that there is no analog to Theorem 4.10 for maximum recoveries. This is another advantage of extended recoveries over recoveries.

If $\mathcal{M}$ is a schema mapping, define a *strong maximum recovery* of $\mathcal{M}$ to be a schema mapping $\mathcal{M}'$ such that (1) $\mathcal{M}'$ is a recovery of $\mathcal{M}$, and (2) for every recovery $\mathcal{M}''$ of $\mathcal{M}$, we have $\mathcal{M}' \subseteq \mathcal{M}''$. If $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$ and $\mathcal{M}'' = (\mathbf{T}, \mathbf{S}, \Sigma'')$, then saying that $\mathcal{M}' \subseteq \mathcal{M}''$ is the same as saying that Σ' logically implies Σ''. So a strong maximum recovery $\mathcal{M}'$ has a specification Σ' that is strongest possible among all recoveries. We now show that schema mappings with a strong maximum recovery are rare. In particular, no schema mapping specified by a finite set of s-t tgds has a strong maximum recovery.

Theorem 4.11. *A schema mapping has a strong maximum recovery if and only if every source instance has exactly one solution. In particular, no schema mapping specified by a finite set of s-t tgds has a strong maximum recovery. These results hold whether we restrict source instances to being ground or not.*

Proof. The proof is the same whether we restrict source instances to being ground or not.

Assume first that every source instance I has exactly one solution J_I. Then it is easy to see that there is a unique recovery, which consists of all pairs (J_I, I). Since this recovery is unique, it is of course a strong maximum recovery.

Assume now that the schema mapping $\mathcal{M}$ has a strong maximum recovery $\mathcal{M}'$; we shall show that every source instance has exactly one solution with respect to $\mathcal{M}$. If some source instance I_0 has no solution, then clearly $(I_0, I_0) \notin \mathcal{M} \circ \mathcal{M}'$, and so $\mathcal{M}'$ is not a recovery, which is a contradiction. Assume now that there is some source instance I_0 with more than one solution; we shall derive a contradiction. Let J_1 and J_2 be distinct solutions for I_0. Define the schema mapping $\mathcal{M}''_1$ as follows. If $I \neq I_0$, then $(J, I) \in \mathcal{M}''_1$ if and only if $(J, I) \in \mathcal{M}'$. If $I = I_0$, then $(J, I) \in \mathcal{M}''_1$ if and only if $J = J_1$. We now show that $\mathcal{M}''_1$ is a recovery of $\mathcal{M}$. First, $(I_0, I_0) \in \mathcal{M} \circ \mathcal{M}''_1$, since $(I_0, J_1) \in \mathcal{M}$ and $(J_1, I_0) \in \mathcal{M}''_1$. Further, if $I \neq I_0$, we know (since $\mathcal{M}'$ is a recovery) that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$, and so there is J such that $(I, J) \in \mathcal{M}$ and $(J, I) \in \mathcal{M}'$. But (J, I) is also in $\mathcal{M}''_1$, and so $(I, I) \in \mathcal{M} \circ \mathcal{M}''_1$. So $\mathcal{M}''_1$ is indeed a recovery of $\mathcal{M}$. Now define the schema mapping $\mathcal{M}''_2$ analogously (but using J_2 instead of J_1), as follows. If $I \neq I_0$, then $(J, I) \in \mathcal{M}''_2$ if and only if $(J, I) \in \mathcal{M}'$. If $I = I_0$, then $(J, I) \in \mathcal{M}''_1$ if and only if $J = J_2$. By an analogous argument to that we used to show that $\mathcal{M}''_1$ is a recovery of $\mathcal{M}$, we have that $\mathcal{M}''_2$ is a recovery of $\mathcal{M}$. Since $\mathcal{M}'$ is a strong maximum recovery of $\mathcal{M}$, we have $\mathcal{M}' \subseteq \mathcal{M}''_1$ and $\mathcal{M}' \subseteq \mathcal{M}''_2$. Hence, $\mathcal{M}' \subseteq \mathcal{M}''_1 \cap \mathcal{M}''_2$. But the only J with $(J, I_0) \in \mathcal{M}''_1$ is J_1, and the only

J with $(J, I_0) \in \mathcal{M}''_2$ is J_2. Hence, there is no J with $(J, I_0) \in \mathcal{M}''_1 \cap \mathcal{M}''_2$, and so since $\mathcal{M}' \subseteq \mathcal{M}''_1 \cap \mathcal{M}''_2$, there is no J with $(J, I_0) \in \mathcal{M}'$. Therefore, $(I_0, I_0) \notin \mathcal{M} \circ \mathcal{M}'$. It follows that $\mathcal{M}'$ is not a recovery of $\mathcal{M}$, which is our desired contradiction.

As for the "in particular," let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. Then no source instance I has a unique solution J, as we now show. Let F be a fact not in J (such a fact F exists since J is finite). Then $J \cup \{F\}$ is also a solution for I. □

Our next main result is Theorem 4.14, which shows that the quantity $\mathcal{C}_{\mathcal{M}} = e(\mathcal{M}) \circ e(\mathcal{M}')$, where $\mathcal{M}'$ is a maximum extended recovery, coincides with $\rightarrow_{\mathcal{M}}$. The proof of Theorem 4.14 makes use of Proposition 4.12 and Lemma 4.13.

PROPOSITION 4.12. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. Then $\rightarrow_{\mathcal{M}} = \rightarrow \circ \rightarrow_{\mathcal{M}} \circ \rightarrow$.*

PROOF. The containment $\rightarrow_{\mathcal{M}} \subseteq \rightarrow \circ \rightarrow_{\mathcal{M}} \circ \rightarrow$ is obvious. The reverse inclusion can be shown by observing that $\rightarrow \subseteq \rightarrow_{\mathcal{M}}$. □

LEMMA 4.13. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. If $\mathcal{M}^* = \{(chase_{\mathcal{M}}(I), I) \mid I$ is a source instance $\}$, then $e(\mathcal{M}) \circ e(\mathcal{M}^*) \ = \rightarrow_{\mathcal{M}}$.*

PROOF. We first show that $\rightarrow_{\mathcal{M}} \subseteq \mathcal{M} \circ \rightarrow \circ \mathcal{M}^*$. Assume that $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$. Then we have $(I_1, chase_{\mathcal{M}}(I_1)) \in \mathcal{M}$, $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$, and $(chase_{\mathcal{M}}(I_2), I_2) \in \mathcal{M}^*$; hence, $(I_1, I_2) \in \mathcal{M}\circ \rightarrow \circ\mathcal{M}^*$.

We now show that $\mathcal{M} \circ \rightarrow \circ \mathcal{M}^* \subseteq \rightarrow_{\mathcal{M}}$. If $(I_1, I_2) \in \mathcal{M}\circ \rightarrow \circ\mathcal{M}^*$, then there is a target instance J such that $(I_1, J) \in \mathcal{M}$ and $J \rightarrow chase_{\mathcal{M}}(I_2)$. Since $chase_{\mathcal{M}}(I_1)$ is a universal solution for I_1, we have that $chase_{\mathcal{M}}(I_1) \rightarrow J$; hence, $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$.

We have shown that $\mathcal{M}\circ \rightarrow \circ\mathcal{M}^* = \rightarrow_{\mathcal{M}}$. So by Proposition 4.12, we have $e(\mathcal{M}) \circ e(\mathcal{M}^*) = \rightarrow_{\mathcal{M}}$. □

THEOREM 4.14. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. The following statements are equivalent:*

(1) *$\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$.*
(2) *$e(\mathcal{M}) \circ e(\mathcal{M}') \ = \rightarrow_{\mathcal{M}}$.*

PROOF. Assume first that $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$. By Theorem 4.10, we know that $\mathcal{M}^*$ is a maximum extended recovery of $\mathcal{M}$, and so $e(\mathcal{M}) \circ e(\mathcal{M}') = e(\mathcal{M})\circ e(\mathcal{M}^*)$. But by Lemma 4.13, we have that $e(\mathcal{M})\circ e(\mathcal{M}^*) \ = \rightarrow_{\mathcal{M}}$. So $e(\mathcal{M})\circ e(\mathcal{M}') \ = \rightarrow_{\mathcal{M}}$, as desired.

Assume now that $e(\mathcal{M}) \circ e(\mathcal{M}') \ = \rightarrow_{\mathcal{M}}$. Then $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$, since $(I, I) \in\rightarrow_{\mathcal{M}}$, and so $(I, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. Let $\mathcal{M}''$ be an arbitrary extended recovery. Since $\mathcal{M}^*$ is a maximum extended recovery, we have $e(\mathcal{M})\circ e(\mathcal{M}^*) \subseteq e(\mathcal{M})\circ e(\mathcal{M}'')$. So by Lemma 4.13, we have $\rightarrow_{\mathcal{M}}\subseteq e(\mathcal{M})\circ e(\mathcal{M}'')$. Therefore, $e(\mathcal{M})\circ e(\mathcal{M}') \subseteq e(\mathcal{M})\circ e(\mathcal{M}'')$. Since $\mathcal{M}''$ is an arbitrary extended recovery, this implies that $\mathcal{M}'$ is a maximum extended recovery. □

The preceding results yield a characterization of the information loss of a schema mapping specified by s-t tgds.

COROLLARY 4.15. *If $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds, then the information loss of $\mathcal{M}$ is $\rightarrow_{\mathcal{M}} \setminus \rightarrow$. In other words, for every maximum extended recovery $\mathcal{M}'$ of $\mathcal{M}$, we have that $(e(\mathcal{M}) \circ e(\mathcal{M}')) \setminus \rightarrow = \rightarrow_{\mathcal{M}} \setminus \rightarrow$.*

PROOF. The proof is immediate from the definition of information loss (Definition 4.5) and Theorem 4.14. □

Note that if schema mapping $\mathcal{M}$ specified by s-t tgds is extended invertible, then $\rightarrow_{\mathcal{M}} = e(\text{Id}) = \rightarrow$, and hence the information loss is empty.

COROLLARY 4.16. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. The following are equivalent:*

(1) *$\mathcal{M}$ is extended invertible.*
(2) $\rightarrow_{\mathcal{M}} = \rightarrow$.
(3) *$\mathcal{M}$ has no information loss, that is, $\rightarrow_{\mathcal{M}} \setminus \rightarrow \; = \; \emptyset$.*

PROOF. It is always true (for schema mappings specified by s-t tgds) that $\rightarrow \subseteq \rightarrow_{\mathcal{M}}$. It follows from Proposition 4.7 that the reverse inclusion $\rightarrow_{\mathcal{M}} \subseteq \rightarrow$ is equivalent to the homomorphism property. So the equivalence of (1) and (2) follows from Theorem 3.15. The equivalence of (2) and (3) is immediate from the definition of information loss. Note that information loss is defined only for schema mappings that have a maximum extended recovery, but we know from Theorem 4.10 that $\mathcal{M}$ has a maximum extended recovery. □

The final result of this section relates extended inverses to maximum extended recoveries.

PROPOSITION 4.17. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds and let $\mathcal{M}'$ be an arbitrary schema mapping. If $\mathcal{M}$ is extended invertible, then $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$ if and only if $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$.*

PROOF. Assume that $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$. Then $e(\mathcal{M}) \circ e(\mathcal{M}') = e(\text{Id})$. Since $(I, I) \in e(\text{Id})$, it follows that $(I, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$, and so $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$. Since $e(\text{Id}) \subseteq \text{e}(\mathcal{M}) \circ \text{e}(\mathcal{M}'')$ for every recovery $\mathcal{M}''$, it follows that $\mathcal{M}'$ is a maximum extended recovery. Note that we did not use yet the assumption that $\mathcal{M}$ is specified by a finite set of s-t tgds.

Assume now that $\mathcal{M}$ is extended invertible, and $\mathcal{M}'$ is an extended maximum recovery of $\mathcal{M}$. By Theorem 4.14, we have that $e(\mathcal{M}) \circ e(\mathcal{M}') = \rightarrow_{\mathcal{M}}$. By Corollary 4.16, we have that $\rightarrow_{\mathcal{M}'} = \rightarrow$. So $e(\mathcal{M}) \circ e(\mathcal{M}') = \rightarrow$. Since $\rightarrow \; = \; e(\text{Id})$, it follows that $e(\mathcal{M}) \circ e(\mathcal{M}') = e(\text{Id})$. Therefore, $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$. □

4.2. Information Loss on Ground Instances

We now consider schema mappings specified by a finite set of s-t tgds but restrict the source instances to be ground. As shown in Arenas et al. [2009], every such schema mapping $\mathcal{M}$ has a maximum recovery $\mathcal{M}'$. By the definition of a maximum recovery, the quantity $\mathcal{M} \circ \mathcal{M}'$ is a constant that depends only on $\mathcal{M}$. This motivates the following notion.

Definition 4.18. Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. The *information loss of $\mathcal{M}$ on ground instances* is the set difference $(\mathcal{M} \circ \mathcal{M}') \setminus \text{Id}$, where $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$ and Id is the identity schema mapping on ground instances.

Intuitively, the information loss of $\mathcal{M}$ measures how much $\mathcal{M}$ deviates from being an invertible mapping. Next, we introduce a notion that is an adaptation to ground instances of the earlier notion of $\rightarrow_{\mathcal{M}}$, and use it to characterize the information loss of $\mathcal{M}$ on ground instances.

Definition 4.19. Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. We write $\rightarrow_{\mathcal{M},g}$ to denote the following schema mapping:

$$\{(I_1, I_2) \mid I_1, I_2 \text{ are ground instances and } \text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)\}.$$

We are now ready to state a few properties of maximum recoveries.

PROPOSITION 4.20. *Let $\mathcal{M}$ be a schema mapping. If $\mathcal{M}'$ is a recovery of $\mathcal{M}$, then $\rightarrow_{\mathcal{M},g} \subseteq \mathcal{M} \circ \mathcal{M}'$.*

PROOF. Let I_1 and I_2 ground instances such that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. Since $\mathcal{M}'$ is a recovery of $\mathcal{M}$, we have that $(I_2, I_2) \in \mathcal{M} \circ \mathcal{M}'$. This means that there is a J such that $(I_2, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$. Hence, J is a solution for I_2 with respect to $\mathcal{M}$, so the hypothesis implies that J must also be a solution for I_1 with respect to $\mathcal{M}$. Thus, we have that $(I_1, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$, and so $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$. □

Definition 4.21. If $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds, then we write $\mathcal{M}_g^*$ to denote the schema mapping $\mathcal{M}_g^* = \{(\mathit{chase}_{\mathcal{M}}(I), I) \mid I$ is a ground instance$\}$.

PROPOSITION 4.22. *If $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds, then $\mathcal{M}_g^*$ is a recovery of $\mathcal{M}$.*

PROOF. For every ground instance I, we have that $(I, I) \in \mathcal{M} \circ \mathcal{M}_g^*$. This is because $(I, \mathit{chase}_{\mathcal{M}}(I)) \in \mathcal{M}$ and $(\mathit{chase}_{\mathcal{M}}(I), I) \in \mathcal{M}_g^*$. □

PROPOSITION 4.23. *If $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds, then*

$$\rightarrow_{\mathcal{M},g} \; = \; \{(I_1, I_2) \mid I_1, I_2 \textit{ are ground instances and } \mathit{chase}_{\mathcal{M}}(I_1) \rightarrow \mathit{chase}_{\mathcal{M}}(I_2)\}.$$

PROOF. Let I_1 and I_2 be ground instances such that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. It follows that $\mathit{chase}_{\mathcal{M}}(I_2) \in \text{Sol}_{\mathcal{M}}(I_1)$. Hence, since $\mathit{chase}_{\mathcal{M}}(I_1)$ is a universal solution for I_1, we have that $\mathit{chase}_{\mathcal{M}}(I_1) \rightarrow \mathit{chase}_{\mathcal{M}}(I_2)$. This shows that:

$$\rightarrow_{\mathcal{M},g} \; \subseteq \; \{(I_1, I_2) \mid I_1, I_2 \text{ are ground instances and } \mathit{chase}_{\mathcal{M}}(I_1) \rightarrow \mathit{chase}_{\mathcal{M}}(I_2)\}.$$

For the other direction, assume that I_1 and I_2 are ground instances such that $\mathit{chase}_{\mathcal{M}}(I_1) \rightarrow \mathit{chase}_{\mathcal{M}}(I_2)$. Let J be a solution for I_2. Hence, $\mathit{chase}_{\mathcal{M}}(I_2) \rightarrow J$, which implies that $\mathit{chase}_{\mathcal{M}}(I_1) \rightarrow J$. Since $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds, and I_1 is ground, it follows that J is a solution for I_1. This shows that $\{(I_1, I_2) \mid I_1, I_2$ are ground instances and $\mathit{chase}_{\mathcal{M}}(I_1) \rightarrow \mathit{chase}_{\mathcal{M}}(I_2)\} \subseteq \rightarrow_{\mathcal{M},g}$. □

PROPOSITION 4.24. *If $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds, then*

$$\rightarrow_{\mathcal{M},g} \; = \; \mathcal{M} \circ \mathcal{M}_g^*.$$

PROOF. It follows from Proposition 4.20 and Proposition 4.22 that $\rightarrow_{\mathcal{M},g} \subseteq \mathcal{M} \circ \mathcal{M}_g^*$.

For the other direction, assume that I_1, I_2 are ground instances such that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}_g^*$. It follows that there is a target instance J such that $(I_1, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}_g^*$. By Definition 4.21, it must be that $J = \mathit{chase}_{\mathcal{M}}(I_2)$, which means that $(I_1, \mathit{chase}_{\mathcal{M}}(I_2)) \in \mathcal{M}$. In turn, this implies that $\mathit{chase}_{\mathcal{M}}(I_1) \rightarrow \mathit{chase}_{\mathcal{M}}(I_2)$, hence, by Proposition 4.23, we have that $(I_1, I_2) \in \rightarrow_{\mathcal{M},g}$. This shows that $\mathcal{M} \circ \mathcal{M}_g^* \subseteq \rightarrow_{\mathcal{M},g}$. □

PROPOSITION 4.25. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. If $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$, then $\mathcal{M} \circ \mathcal{M}' = \; \rightarrow_{\mathcal{M},g}$. Consequently, the information loss of $\mathcal{M}$ on ground instances is equal to $\rightarrow_{\mathcal{M},g} \; \setminus \text{Id}$.*

PROOF. By Proposition 4.20, we have that $\rightarrow_{\mathcal{M},g} \subseteq \; \mathcal{M} \circ \mathcal{M}'$. By Proposition 4.22, we have that $\mathcal{M}_g^*$ is a recovery of $\mathcal{M}$, hence $\mathcal{M} \circ \mathcal{M}' \subseteq \mathcal{M} \circ \mathcal{M}_g^*$. Since, by Proposition 4.24, $\mathcal{M} \circ \mathcal{M}_g^* = \rightarrow_{\mathcal{M},g}$, we have that $\mathcal{M} \circ \mathcal{M}' \subseteq \rightarrow_{\mathcal{M},g}$. □

We note that the preceding proposition can be obtained from results in the full version of Arenas et al. [2009], which however, does not have the notion of information loss.

4.3. An Alternative Notion of Extended Recovery

To cope with nulls in source instances, we replaced the notion of a solution of a source instance with respect to a schema mapping $\mathcal{M}$ by that of an extended solution, and then worked with the homomorphic extension $e(\mathcal{M})$ of $\mathcal{M}$. Furthermore, we obtained the notions of extended recovery and maximum extended recovery from the notions of recovery and maximum recovery introduced in Arenas et al. [2009] by replacing not only $\mathcal{M}$ by $e(\mathcal{M})$, but also replacing every mention of a schema mapping $\mathcal{N}$ in the definition of each of these two notions by its homomorphic extension $e(\mathcal{N})$. In particular, the composition $\mathcal{M} \circ \mathcal{M}'$ of two schema mappings $\mathcal{M}$ and $\mathcal{M}'$ was replaced by the composition $e(\mathcal{M}) \circ e(\mathcal{M}')$ of their homomorphic extensions.

Our approach of coping with nulls in source instances can also be construed as changing the semantics of satisfaction of dependencies so that extended solutions are used in place of solutions.[2] In turn, this suggests an alternative notion of extended recovery and a corresponding alternative notion of maximum extended recovery. Specifically, given a schema mapping $\mathcal{M}$, we consider the notions of a recovery of $e(\mathcal{M})$ and of a maximum recovery of $e(\mathcal{M})$. In other words, we use the notions of recovery and maximum recovery introduced in Arenas et al. [2009], but apply them directly to the homomorphic extension $e(\mathcal{M})$ of a schema mapping $\mathcal{M}$, where source instances may contain nulls. We show that on the one hand these alternative notions differ from those of extended recovery and maximum extended recovery, but on the other hand they give rise to the same information loss for schema mappings specified by a finite set of s-t tgds.

Before stating and proving the results of this section, we spell out the meaning of the alternative notions we study here. Let $\mathcal{M}$ be a schema mapping. A schema mapping $\mathcal{M}'$ is a recovery of $e(\mathcal{M})$ if $(I, I) \in e(\mathcal{M}) \circ \mathcal{M}'$, for every source instance I. A schema mapping $\mathcal{M}'$ is a maximum recovery of $e(\mathcal{M})$ if (1) $\mathcal{M}'$ is a recovery of $e(\mathcal{M})$, and (2) for every recovery $\mathcal{M}''$ of $e(\mathcal{M})$, we have $e(\mathcal{M}) \circ \mathcal{M}' \subseteq e(\mathcal{M}) \circ \mathcal{M}''$. Thus, in these notions, only the schema mapping $\mathcal{M}$ is replaced by its homomorphic extension $e(\mathcal{M})$.

Proposition 4.26. *Let $\mathcal{M}$ be a schema mapping.*

(1) *If $\mathcal{M}'$ is a recovery of $e(\mathcal{M})$, then $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$.*
(2) *If $\mathcal{M}'$ is a maximum recovery of $e(\mathcal{M})$, then $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$.*

Proof. The first part follows from the definitions and the fact that $\mathcal{M}' \subseteq e(\mathcal{M}')$, for every schema mapping $\mathcal{M}'$. For the second part, assume that $\mathcal{M}'$ is a maximum recovery of $e(\mathcal{M})$. By the first part, we know that $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$. So, it remains to show that if $\mathcal{M}''$ is an extended recovery of $\mathcal{M}$, then $e(\mathcal{M}) \circ e(\mathcal{M}') \subseteq e(\mathcal{M}) \circ e(\mathcal{M}'')$. Since M'' is an extended recovery of $\mathcal{M}$, we have that $e(\mathcal{M}'')$ is a recovery of $e(\mathcal{M})$, hence $e(\mathcal{M}) \circ \mathcal{M}' \subseteq e(\mathcal{M}) \circ e(\mathcal{M}'')$ (since $\mathcal{M}'$ is a maximum recovery of $e(\mathcal{M})$). It follows that $e(\mathcal{M}) \circ \mathcal{M}' \circ \rightarrow \subseteq e(\mathcal{M}) \circ e(\mathcal{M}'') \circ \rightarrow$, which in turn, easily implies that $e(\mathcal{M}) \circ e(\mathcal{M}') \subseteq e(\mathcal{M}) \circ e(\mathcal{M}'')$. □

The next result reveals that neither converse of Proposition 4.26 is true.

Proposition 4.27. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be the schema mapping with $\Sigma = \{P(x, y) \rightarrow \exists z(Q(x, z) \wedge Q(z, y))\}$, and let $\mathcal{M}'$ be the schema mapping $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$ with $\Sigma' = \{Q(x, z) \wedge Q(z, y) \rightarrow P(x, y)\}$. Then $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$ (in fact, it is an extended inverse of $\mathcal{M}$), but it is not a recovery of $e(\mathcal{M})$, and so it is not a maximum recovery of $e(\mathcal{M})$.*

[2] This viewpoint was pointed out to us by M. Arenas, J. Pérez, J. Reutter, and C. Riveros in a private communication.

Proof. The schema mapping $\mathcal{M}'$ was shown to be an extended inverse of $\mathcal{M}$ in Example 3.23; consequently, by Proposition 4.17, $\mathcal{M}'$ is also a maximum extended recovery of $\mathcal{M}$. We will show that $\mathcal{M}'$ is not a recovery of $e(\mathcal{M})$ by exhibiting a source instance I such that $(I, I) \notin e(\mathcal{M}) \circ \mathcal{M}'$. For this, consider the ground source instance $I = \{P(a, b), P(b, c)\}$. Towards a contradiction, assume that $(I, I) \in e(\mathcal{M}) \circ \mathcal{M}'$. Then there is a source instance I' and two target instances J and J' such that $I \to I'$, $(I', J') \in \mathcal{M}$, $J' \to J$, and $(J, I) \in \mathcal{M}'$. Since a, b, and c are constants, it follows that J must contain the facts $Q(a, U)$, $Q(U, b)$, $Q(b, V)$, $Q(V, c)$, for some values (constants or nulls) U and V. Furthermore, since $(J, I) \in \mathcal{M}'$, we must have that $P(U, V) \in I$, which implies that either $U = a$ and $V = b$ or $U = b$ and $V = c$. If $U = a$, then J contains the fact $Q(a, a)$, and so I must contain the fact $P(a, a)$, which is false. If $U = b$, then J contains the fact $Q(b, b)$, and so I must contain the fact $P(b, b)$, which is false as well. We conclude that $(I, I) \notin e(\mathcal{M}) \circ \mathcal{M}'$ and, consequently, $\mathcal{M}'$ is not a recovery of $e(\mathcal{M})$. □

Propositions 4.26 and 4.27 imply that the notions of a recovery of $e(\mathcal{M})$ and a maximum recovery of $e(\mathcal{M})$ are stricter notions than those of an extended recovery of $\mathcal{M}$ and a maximum extended recovery of $\mathcal{M}$, respectively. Note that Proposition 4.27 exhibits a natural extended recovery $\mathcal{M}'$ of $\mathcal{M}$ that has good properties for reverse data exchange ($\mathcal{M}'$ is a chase-inverse), whereas $\mathcal{M}'$ is not a maximum recovery of $e(\mathcal{M})$.

Now, suppose that $\mathcal{M}$ is a schema mapping, $\mathcal{M}'$ is a maximum recovery of $e(\mathcal{M})$, and $\mathcal{M}''$ is a maximum extended recovery of $\mathcal{M}$. From the definitions of these notions, it follows that the quantities $e(\mathcal{M}) \circ \mathcal{M}'$ and $e(\mathcal{M}) \circ e(\mathcal{M}'')$ depend only on $\mathcal{M}$; in particular, they are independent of the actual choices of $\mathcal{M}'$ and $\mathcal{M}''$ as a maximum recovery of $e(\mathcal{M})$ and a maximum extended recovery of $\mathcal{M}$, respectively. Furthermore, since $e(\mathcal{M}'')$ is also a recovery of $e(\mathcal{M})$, we have that $e(\mathcal{M}) \circ \mathcal{M}' \subseteq e(\mathcal{M}) \circ e(\mathcal{M}'')$. Thus, at first sight, it appears that the notion of maximum recovery of $e(\mathcal{M})$ may lead into a new notion of information loss of a schema mapping $\mathcal{M}$ (namely, the set difference $(e(\mathcal{M}) \circ \mathcal{M}') \setminus \to$, where $\mathcal{M}'$ is a maximum recovery of $e(\mathcal{M})$) that could be strictly smaller than the notion of information loss arising from the notion of maximum extended recovery. Our next result, however, implies that, for schema mappings specified by a finite set of s-t tgds, this is not the case.

Theorem 4.28. *If $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds, then the schema mapping $\mathcal{M}^* = \{ (chase_{\mathcal{M}}(I), I) \mid I$ is a source instance $\}$ is a maximum recovery of $e(\mathcal{M})$. Furthermore, we have that $e(\mathcal{M}) \circ \mathcal{M}^* = e(\mathcal{M}) \circ e(\mathcal{M}^*)$.*

Proof. First, $\mathcal{M}^*$ is a recovery of $e(\mathcal{M})$ because $(I, I) \in \mathcal{M} \circ \mathcal{M}^* \subseteq e(\mathcal{M}) \circ \mathcal{M}^*$. Suppose now that $\mathcal{M}'$ is a recovery of $e(\mathcal{M})$. We will show that $e(\mathcal{M}) \circ \mathcal{M}^* \subseteq e(\mathcal{M}) \circ \mathcal{M}'$. Actually, we will show a stronger statement, namely, that $\mathcal{M}^* \subseteq \to \circ \mathcal{M}'$, which implies that $e(\mathcal{M}) \circ \mathcal{M}^* \subseteq e(\mathcal{M}) \circ \to \circ \mathcal{M}' = e(\mathcal{M}) \circ \mathcal{M}'$. Let I be an arbitrary source instance. Since $\mathcal{M}'$ is a recovery of $e(\mathcal{M})$, we have that $(I, I) \in e(\mathcal{M}) \circ \mathcal{M}'$. It follows that there is a target instance J such that $(I, J) \in e(\mathcal{M})$ and $(J, I) \in \mathcal{M}'$. Since J is an extended solution for I with respect to $\mathcal{M}$ and since $chase_{\mathcal{M}}(I)$ is a universal extended solution for I with respect to $\mathcal{M}$, we have that $chase_{\mathcal{M}}(I) \to J$, and so $(chase_{\mathcal{M}}(I), I) \in \to \circ \mathcal{M}'$. This completes the proof that $\mathcal{M}^* \subseteq \to \circ \mathcal{M}'$, which in turn implies that $\mathcal{M}^*$ is a maximum recovery of $e(\mathcal{M})$.

Next we show that $e(\mathcal{M}) \circ \mathcal{M}^* \subseteq e(\mathcal{M}) \circ e(\mathcal{M}^*)$. To begin with, since $\mathcal{M}^* \subseteq e(\mathcal{M}^*)$, we have that $e(\mathcal{M}) \circ \mathcal{M}^* \subseteq e(\mathcal{M}) \circ e(\mathcal{M}^*)$. For the other containment, we will show that $e(\mathcal{M}) \circ \mathcal{M}^* \circ \to \subseteq e(\mathcal{M}) \circ \mathcal{M}^*$, which easily implies that $e(\mathcal{M}) \circ e(\mathcal{M}^*) \subseteq e(\mathcal{M}) \circ \mathcal{M}^*$. Let I, I', and I'' be source instances such that $(I, I') \in e(\mathcal{M}) \circ \mathcal{M}^*$ and $I' \to I''$. We have to show that $(I, I'') \in e(\mathcal{M}) \circ \mathcal{M}^*$. Let K be a target instance such that $(I, K) \in e(\mathcal{M})$ and $(K, I') \in M^*$. Then $K = chase_{\mathcal{M}}(I')$ and so we have that $(I, chase_{\mathcal{M}}(I')) \in e(\mathcal{M})$ and $(chase_{\mathcal{M}}(I'), I') \in \mathcal{M}^*$. Since $I' \to I''$, we also have that $chase_{\mathcal{M}}(I') \to chase_{\mathcal{M}}(I'')$.

Consequently, $(I, chase_{\mathcal{M}}(I'')) \in e(\mathcal{M})$ and $(chase_{\mathcal{M}}(I''), I'') \in \mathcal{M}^*$, which implies that $(I, I'') \in e(\mathcal{M}) \circ \mathcal{M}^*$. This completes the proof that $e(\mathcal{M}) \circ \mathcal{M}^* \circ \rightarrow \subseteq e(\mathcal{M}) \circ \mathcal{M}^*$ and the proof of the theorem. □

COROLLARY 4.29. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. If $\mathcal{M}'$ is a maximum recovery of $e(\mathcal{M})$ and $\mathcal{M}''$ is a maximum extended recovery of $\mathcal{M}$, then $e(\mathcal{M}) \circ \mathcal{M}' = e(\mathcal{M}) \circ e(\mathcal{M}'')$.*

PROOF. This follows from Theorem 4.28 and the fact that $\mathcal{M}^*$ is not only a maximum recovery of $e(\mathcal{M})$, but also (by Theorem 4.10) a maximum extended recovery of $\mathcal{M}$. □

Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds. The final result of this section sheds additional light on the relationship between maximum extended recoveries of $\mathcal{M}$ and maximum recoveries of $e(\mathcal{M})$.

COROLLARY 4.30. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds and let $\mathcal{M}'$ be a recovery of $e(\mathcal{M})$. Then the following statements are equivalent.*

(1) *$\mathcal{M}'$ is a maximum recovery of $e(\mathcal{M})$.*
(2) *$\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$.*

PROOF. We have that (1) $\Rightarrow$ (2) by part (2) of Proposition 4.26. We now show that (2) $\Rightarrow$ (1). Clearly, we have that $e(\mathcal{M}) \circ \mathcal{M}' \subseteq e(\mathcal{M}) \circ e(\mathcal{M}')$. Since by assumption $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$, and since, by Theorem 4.10 we know that $\mathcal{M}^*$ is also a maximum extended recovery of $\mathcal{M}$, it follows that $e(\mathcal{M}) \circ e(\mathcal{M}') = e(\mathcal{M}) \circ e(\mathcal{M}^*)$. Now $e(\mathcal{M}) \circ e(\mathcal{M}^*) = e(\mathcal{M}) \circ \mathcal{M}^*$, by Theorem 4.28. Putting together what we have shown, we have that $e(\mathcal{M}) \circ \mathcal{M}' \subseteq e(\mathcal{M}) \circ \mathcal{M}^*$. Therefore, since $\mathcal{M}^*$ is a maximum recovery of $e(\mathcal{M})$, we have that $\mathcal{M}'$ is a maximum recovery of $e(\mathcal{M})$. □

4.4. Relationship to Query Answering in Incomplete Databases

As it has become quite clear by now, the notion of an extended solution (Definition 3.1) has been central to our study of reverse data exchange. This notion can also be construed as an alternative semantics of satisfaction of an s-t tgd ψ by a pair (I, J) of a source instance I and a target instance J, where both I and J may contain labeled nulls. In effect, we have replaced the standard notion $(I, J) \models \psi$ of satisfaction by a notion of *extended satisfaction* $(I, J) \models_e \psi$, where $(I, J) \models_e \psi$ means that there exist instances I' and J' such that $I \rightarrow I'$, $(I', J') \models \psi$, and $J' \rightarrow J$. In symbols, we have that $\models_e$ is $\rightarrow \circ \models \circ \rightarrow$.

There is a large body of work in the literature addressing the semantics of queries (and, in particular, of first-order formulas) when the instances are allowed to contain labeled nulls. In this section, we examine the relationship between this work and the notion of extended satisfaction that we considered here.

The papers Imieliński and Lipski [1983, 1984] have become a standard reference for the semantics of queries posed over *incomplete databases*, that is databases in which facts may contain labeled nulls as values. Assume that $\mathbf{R}$ is a schema and K is an instance over $\mathbf{R}$ whose facts contain values from $\underline{\mathsf{Const}} \cup \underline{\mathsf{Var}}$. Imieliński and Lipski argued that K represents an infinite set $\text{Rep}(K)$ of ground instances over $\mathbf{R}$, where a ground instance K' is a member of $\text{Rep}(K)$ if and only if $K \rightarrow K'$. They then went on to define the semantics of a query q on K as the intersection $\bigcap\{q(K') : K' \in \text{Rep}(K)\}$. In other words, if we let $\models_{\text{IL}}$ denote the Imieliński-Lipski semantics, then for a first-order sentence ψ, we have that $K \models_{\text{IL}} \psi$ if and only if $K' \models \psi$, for every ground instance $K' \in \text{Rep}(K)$. Imieliński and Lipski were mainly concerned with answering unions of conjunctive queries over incomplete databases, in which case their semantics is nontrivial and, in fact, interesting. On the other hand, this semantics can be quite trivial when applied to s-t tgds. For example, consider the unary copy s-t tgd $P(x) \rightarrow P'(x)$. Let I be an arbitrary source instance and J an arbitrary target instance. Then there are ground instances I' and J' such that $(I', J') \in \text{Rep}((I, J))$, but $(I', J') \not\models P(x) \rightarrow P'(x)$. To see this, take ground instances I'' and J'' obtained from I and J, respectively, by replacing each labeled null by a distinct constant, and let I' be the ground instance obtained from I'' by adding a ground fact $P(a)$ such that $P'(a)$ does not occur in J''; finally, put $J' = J''$. Then, it is indeed the case that $(I', J') \in \text{Rep}((I, J))$, but $(I', J') \not\models P(x) \rightarrow P'(x)$. Consequently, for every source instance I and every target instance J, we have that $(I, J) \not\models_{\text{IL}} P(x) \rightarrow P'(x)$. In general, the notion of $\models_{\text{IL}}$ satisfaction gives rise to trivial semantics for those s-t tgds that export values from the source to the target.

More recently, Afrati, Li, and Pavlaki [Afrati et al. 2008] studied the semantics and the complexity of query answering in the context of data exchange with incomplete source instances. As a matter of fact, they considered two flavors of *certain answers* semantics, namely $certain^{A}_{\mathcal{M}}(q, I)$ and $certain^{B}_{\mathcal{M}}(q, I)$, where $\mathcal{M}$ is a schema mapping specified by s-t tgds, source tgds, and target egds and tgds, q is a query over the target schema, and I is a source instance that may contain labeled nulls. Note that for schema mappings $\mathcal{M}$ specified by just s-t tgds (which is our concern here), these two notions of certain answers coincide; thus, in what follows, we will refer to just one of them, say, to $certain^{B}_{\mathcal{M}}(q, I)$. To define $certain^{B}_{\mathcal{M}}(q, I)$, Afrati, Li, and Pavlaki considered a notion of solution that is related to, yet is different from, our notion of extended solution. Assume that $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a schema mapping such that Σ is a finite set of s-t tgds. Let I be a source instance that may contain nulls. We say that a target instance J is a *B-solution for I with respect to* $\mathcal{M}$ if there is an instance $I' \in \text{Rep}(I)$ such that $(I', J) \models \Sigma$. For a query q over the target schema, Afrati et al. [2008] define $certain^{B}_{\mathcal{M}}(q, I) = \bigcap\{q(J) : J \text{ is a } B\text{-solution for } I \text{ with respect to } \mathcal{M}\}$. In effect, Afrati et al. [2008] replace the standard notion $\models$ of satisfaction by the notion $\models_B$, where $(I, J) \models_B \psi$ if there is an instance $I' \in \text{Rep}(I)$ such that $(I', J) \models \psi$. In symbols, we have that $\models_B$ is $\overset{g}{\rightarrow} \circ \models$, where $\overset{g}{\rightarrow} = \{(I, I') : I' \text{ is ground and } I \rightarrow I'\}$.

Clearly, if $(I, J) \models_B \psi$, then also $(I, J) \models_e \psi$. It is easy to see, however, that the converse need not be true. In other words, every B-solution is an extended solution, but there are extended solutions that are not B-solutions. In fact, B-solutions suffer from a particular limitation that we explain next. As we saw earlier, in Proposition 3.11, if $\mathcal{M}$ is a schema mapping specified by a finite set of s-t tgds, then extended universal solutions always exist, since for every source instance I, we have that $chase_{\mathcal{M}}(I)$ is an extended universal solution for I with respect to $\mathcal{M}$. In contrast, $chase_{\mathcal{M}}(I)$ need not be a B-solution for I, even though it has homomorphisms to every B-solution (since every B-solution is an extended solution); in particular, this holds true even for the schema mapping specified by the unary copy s-t tgd $P(x) \rightarrow P'(x)$, if I is a source instance containing nulls. Furthermore, for this schema mapping, if I is a source instance containing nulls, then no B-solution for I is universal, that is, no B-solution for I has homomorphisms to every B-solution for I.

Finally, as regards the semantics of conjunctive queries over the target schema, there is no difference between the certain answers obtained using extended solutions and those obtained using B-solutions. Specifically, assume that $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a schema mapping such that Σ is a finite set of s-t tgds. If q is a query over the target schema and I is a source instance that may contain nulls, let us put $certain^{e}_{\mathcal{M}}(q, I) = \bigcap\{q(J) : J \text{ is an extended solution for } I \text{ with respect to } \mathcal{M}\}$. It can be shown that if q is a conjunctive query, then $certain^{e}_{\mathcal{M}}(q, I) = certain^{B}_{\mathcal{M}}(q, I)$, for every source instance I. This is so because it can be shown that both $certain^{e}_{\mathcal{M}}(q, I)$ and $certain^{B}_{\mathcal{M}}(q, I)$ are equal to $q(chase_{\mathcal{M}}(I))_{\downarrow}$, where $q(chase_{\mathcal{M}}(I))_{\downarrow}$ is the subset of $q(chase_{\mathcal{M}}(I))$ obtained by removing all facts containing at least one labeled null.

Thus, extended solutions are superior to B-solutions because they posses a smoother theory (e.g., extended universal solutions always exist), while at the same time they give rise to the same notion of certain answers as the one arising from B-solutions.

5. MAXIMUM EXTENDED RECOVERIES: LANGUAGE

The main result of this section is a polynomial-time algorithm that, given a schema mapping $\mathcal{M}$ specified by s-t tgds, produces a maximum extended recovery of $\mathcal{M}$ specified by a formula of existential second-order logic. It is an open problem whether or not every such schema mapping $\mathcal{M}$ has a maximum extended recovery specified by a formula of first-order logic.

We note that Arenas et al. [2009b] gave a polynomial-time algorithm for producing a maximum recovery of $\mathcal{M}$ that is specified in existential second-order logic. It is not obvious whether the algorithm of Arenas et al. [2009b] can be used to obtain maximum extended recoveries. We also note that our algorithm is very different from theirs. However, the fragment of existential second-order logic that we use is very similar to that used in Arenas et al. [2009b]: both involve formulas that are equivalent to formulas of the form $\exists \mathbf{f}(C_1 \wedge \cdots \wedge C_m)$, where $\mathbf{f}$ is a collection of function symbols, each C_i is of the form $\alpha \rightarrow (D_1 \vee \cdots \vee D_n)$ with α a conjunction of atomic formulas including the *Constant* predicate, and with each D_j a conjunction of atomic formulas (including, in our case, also the *Null* predicate), equalities and inequalities of terms.[3] Our construction actually also allows formulas $dom_{\mathbf{U}}$ to appear in α; we now define them and show how we can eliminate them.

Let $\mathbf{U}$ be a schema. In our cases of interest, we shall let $\mathbf{U}$ be either $\mathbf{S}$ (the source schema), $\mathbf{T}$ (the target schema), or $\mathbf{S} \cup \mathbf{T}$ (which consists of those relation symbols in either the source schema or the target schema). The formula $dom_{\mathbf{U}}(\mathbf{x})$ is intended to be an abbreviation for the statement that the members of $\mathbf{x}$ are each members of the active domain of an instance of $\mathbf{U}$. We now show by example how we represent these formulas $dom_{\mathbf{U}}(\mathbf{x})$ in our context, and how we can eliminate them. Assume that $\mathbf{U_1}$ consists of the unary relation symbol P and the binary relation symbol Q, and $\mathbf{U_2}$ consists of the binary relation symbol R. Let y_1 and y_2 be new variables. The formula $\forall \mathbf{x}(dom_{\mathbf{U_1}}(x_1) \wedge dom_{\mathbf{U_2}}(x_2) \wedge \alpha \rightarrow \beta)$ can be taken to be shorthand for the formula:

$$\forall \mathbf{x} \forall y_1 \forall y_2((P(x_1) \vee Q(x_1, y_1) \vee Q(y_1, x_1)) \wedge (R(x_2, y_2) \vee R(y_2, x_2)) \wedge \alpha \rightarrow \beta).$$

This formula is of the form

$$\forall \mathbf{x} \forall y_1 \forall y_2((A_1 \vee A_2 \vee A_3) \wedge (B_1 \vee B_2) \wedge \alpha \rightarrow \beta).$$

We can replace this formula by the six formulas, $\forall \mathbf{x} \forall y_1 \forall y_2(A_i \wedge B_j \wedge \alpha \rightarrow \beta)$, for $1 \leq i \leq 3$ and $1 \leq j \leq 2$.

In the algorithm we shall give shortly, we shall speak of the *Skolemized form* of an s-t tgd. This is obtained by replacing each existentialized variable by a new Skolem function, in the standard way. For example, the Skolemized form of the s-t tgd $Q(x_3) \wedge R(x_3, x_4) \rightarrow \exists x_1 \exists x_2 P(x_1, x_3, x_3, x_2)$ is $Q(x_3) \wedge R(x_3, x_4) \rightarrow P(f_1(x_3, x_4), x_3, x_3, f_2(x_3, x_4))$, where f_i is the Skolem function corresponding to the existentially quantified variable x_i, for $i = 1, 2$. Note that the arguments of f_1 and f_2 are the variables appearing in the premise, namely x_3 and x_4. Note also that the Skolemized form of a full s-t tgd σ is σ itself.

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ is a finite set of s-t tgds. Let

$$\mathcal{M}^{\sharp} = \{(J, I) \mid J \rightarrow chase_{\mathcal{M}}(I) \text{ and } I \text{ is a source instance}\}.$$

[3] Both our construction and the construction in Arenas et al. [2009b] actually take each C_i to be of the form $\exists \mathbf{x}(\alpha \rightarrow (D_1 \vee \cdots \vee D_n))$, but the existential quantifiers can be eliminated by adding extra existentially-quantified function symbols to $\exists \mathbf{f}$.

As before, let

$$\mathcal{M}^* = \{(chase_{\mathcal{M}}(I), I) \mid I \text{ is a source instance}\}.$$

It is easy to see that $\mathcal{M}^\sharp = {\to} \circ \mathcal{M}^*$. Therefore, $e(\mathcal{M}^\sharp) = e(\mathcal{M}^*)$. Hence, since $\mathcal{M}^*$ is a strong maximum extended recovery of $\mathcal{M}$ by Theorem 4.10, it follows that $\mathcal{M}^\sharp$ is a strong maximum extended recovery of $\mathcal{M}$. We now give an algorithm that generates a formula σ' in existential second-order logic that we shall show specifies $\mathcal{M}^\sharp$.

As noted earlier, our existential second-order language for maximum extended recoveries makes use of formulas of the form $Null(x)$, which we define to be simply an abbreviation for $\neg Constant(x)$. Of course, we could instead take *Null* to be a new relation symbol, and define its semantics similarly to how we defined the semantics of *Constant*.

ALGORITHM: MaxExtendedRecovery($\mathcal{M}$)

Input: A schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ is a finite set of s-t tgds.
Output: A schema mapping $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \sigma')$ that is a strong maximum extended recovery of $\mathcal{M}$, where σ' is a formula in existential second-order logic.

1. (*Create a Skolemized and normalized form of* Σ.) Create Σ_1 from Σ by Skolemizing each s-t tgd in Σ. Create Σ_2 from Σ_1 by replacing each member $\alpha \to (\beta_1 \wedge \cdots \wedge \beta_r)$ in Σ_1, where the formulas β_i are atomic, by the formulas $\alpha \to \beta_1, \ldots, \alpha \to \beta_r$, so that the conclusions are singletons.
2. (*Create formulas that describe a homomorphism and Skolem functions.*) Let $\mathbf{f}$ consist of the Skolem function symbols created in the previous step, and let h be a new unary function symbol (that will represent a homomorphism). Let Σ_1' consist of formulas of the following form (where, as usual, we do not bother to write the leading universal quantifiers).
   a. $dom_{\mathbf{T}}(x) \wedge Constant(x) \to (h(x) = x)$
   b. $dom_{\mathbf{S}}(\mathbf{x}) \to Null(f(\mathbf{x}))$, for each f in $\mathbf{f}$
   c. $dom_{\mathbf{S}}(x_1) \wedge \cdots \wedge dom_{\mathbf{S}}(x_n) \wedge dom_{\mathbf{S}}(x_1') \wedge \cdots \wedge dom_{\mathbf{S}}(x_n')$
   $\wedge (f(x_1, \ldots, x_n) = f(x_1', \ldots, x_n')) \to ((x_1 = x_1') \wedge \cdots \wedge (x_n = x_n'))$, for each f in $\mathbf{f}$
   d. $dom_{\mathbf{S}}(\mathbf{x}) \wedge dom_{\mathbf{S}}(\mathbf{x}') \to (f(\mathbf{x}) \neq f'(\mathbf{x}'))$, for distinct f, f' in $\mathbf{f}$
   e. $dom_{\mathbf{S}}(\mathbf{x}) \wedge dom_{\mathbf{S} \cup \mathbf{T}}(y) \to (f(\mathbf{x}) \neq y)$

   Thus, (a) says that h maps each constant to itself, (b) says that the range of each Skolem function consists only of null values, (c) and (d) say that the Skolem functions generate distinct nulls, and (e) says that these values are all new.
3. (*Create formulas about the homomorphism.*) For each relation symbol P of $\mathbf{T}$, if P is k-ary, let $y_1, \ldots, y_k$ be new, distinct variables, and initialize the set S_P to be empty. For each formula $\alpha(\mathbf{x}) \to P(t_1, \ldots, t_k)$ in Σ_2 (where the conclusion contains the relation symbol P), add to S_P the formula $\exists \mathbf{x}(\alpha(\mathbf{x}) \wedge (h(y_1) = t_1) \wedge \cdots \wedge (h(y_k) = t_k))$. Let τ_P be the formula $P(y_1, \ldots, y_k) \to \bigvee \{\phi \mid \phi \in S_P\}$. Let Σ_2' consist of these formulas τ_P (one for each relation symbol P of $\mathbf{T}$). Let σ be the conjunction of the members of $\Sigma_1' \cup \Sigma_2'$, and let σ' be the formula $\exists h \exists \mathbf{f} \sigma$.
   Return $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \sigma')$.

Note that each t_i in Step 3 is either a variable or a term of the form $f(x_{i_1}, \ldots, x_{i_k})$, where $x_{i_1}, \ldots, x_{i_k}$ are variables in $\mathbf{x}$. Since also $\alpha(\mathbf{x})$ implies $dom_{\mathbf{S}}(\mathbf{x})$, it follows that $f(x_{i_1}, \ldots, x_{i_k})$ is the result of applying f with all of its arguments in the active domain of the source instance. This is why it was sufficient to restrict variables in (b), (c), and (d) of Step 2 to be in the active domain of the source instance, and to restrict the variables of $\mathbf{x}$ in (e) of Step 2 to be in the active domain of the source instance.

We now give an example of the application of the algorithm.

Example 5.1. Assume that Σ consists of the following two s-t tgds:

$$\begin{array}{l} A(x_1, x_2) \wedge B(x_1) \rightarrow P(x_1, x_2, x_1, x_1) \\ C(x_1, x_2, x_2, x_3) \rightarrow \exists y (P(x_1, x_2, x_2, y) \wedge Q(y, x_2)). \end{array}$$

Then Σ_1 consists of:

$$\begin{array}{l} A(x_1, x_2) \wedge B(x_1) \rightarrow P(x_1, x_2, x_1, x_1) \\ C(x_1, x_2, x_2, x_3) \rightarrow (P(x_1, x_2, x_2, f(x_1, x_2, x_3)) \wedge Q(f(x_1, x_2, x_3), x_2)). \end{array}$$

We then have Σ_2 consisting of:

$$\begin{array}{l} A(x_1, x_2) \wedge B(x_1) \rightarrow P(x_1, x_2, x_1, x_1) \\ C(x_1, x_2, x_2, x_3) \rightarrow P(x_1, x_2, x_2, f(x_1, x_2, x_3)) \\ C(x_1, x_2, x_2, x_3) \rightarrow Q(f(x_1, x_2, x_3), x_2). \end{array}$$

Then Σ_2' in Step 3 consists of:

$$\begin{array}{rcl} P(y_1, y_2, y_3, y_4) & \rightarrow & (\exists x_1 \exists x_2 (A(x_1, x_2) \wedge B(x_1) \\ & & \wedge (h(y_1) = x_1) \wedge (h(y_2) = x_2) \wedge (h(y_3) = x_1) \wedge (h(y_4) = x_1)) \\ & \vee & \exists x_1 \exists x_2 \exists x_3 (C(x_1, x_2, x_2, x_3) \\ & & \wedge (h(y_1) = x_1) \wedge (h(y_2) = x_2) \wedge (h(y_3) = x_2) \wedge (h(y_4) = f(x_1, x_2, x_3)) \\ Q(y_1, y_2) & \rightarrow & (C(x_1, x_2, x_2, x_3) \wedge (h(y_1) = f(x_1, x_2, x_3)) \wedge (h(y_2) = x_2)). \end{array}$$

As before, let Σ_1' be the conjunction of the formulas in Step 2, and let σ be the conjunction of the members of $\Sigma_1' \cup \Sigma_2'$. Then σ' is the formula $\exists h \exists f \sigma$.

Theorem 5.2. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping where Σ is a finite set of s-t tgds. Then MaxExtendedRecovery($\mathcal{M}$) produces a schema mapping $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \sigma')$ where σ' specifies $\mathcal{M}^\sharp$.*

Proof. Assume first that $(J, I) \in \mathcal{M}^\sharp$, so that $J \rightarrow chase_{\mathcal{M}}(I)$. We now show that $(J, I) \models \sigma'$. We instantiate h by a homomorphism from J to $chase_{\mathcal{M}}(I)$, and we instantiate the members of $\mathbf{f}$ by the Skolem functions that produce the null values that arise in the naive chase.[4] Clearly (J, I) satisfies Σ_1' under our choice of the functions. To conclude the proof that $(J, I) \models \sigma'$, we must show that (J, I) satisfies τ_P for each relation symbol P of $\mathbf{T}$. Assume that $P(y_1, \ldots, y_k)$ holds in J. Since h is a homomorphism from J to $chase_{\mathcal{M}}(I)$, there is some formula $\alpha(\mathbf{x}) \rightarrow P(t_1, \ldots, t_k)$ in Σ_2 (where the conclusion contains the relation symbol P), such that the result of chasing I with $\alpha(\mathbf{x}) \rightarrow P(t_1, \ldots, t_k)$ generates $P(h(y_1), \ldots, h(y_k))$. Thus, there is some choice of $\mathbf{x}$ such that under this choice of $\mathbf{x}$ and for our functions, $\alpha(\mathbf{x})$ holds and $h(y_1) = t_1, \ldots, h(y_k) = t_k$. That is, (J, I) satisfies $\exists \mathbf{x}(\alpha(\mathbf{x}) \wedge (h(y_1) = t_1) \wedge \cdots \wedge (h(y_k) = t_k))$. But τ_P says exactly that one such formula holds. So indeed, (J, I) satisfies τ_P, as desired.

Conversely, assume that $(J, I) \models \sigma'$; we must show that $(J, I) \in \mathcal{M}^\sharp$. It follows from the formulas (b), (c), (d), and (e) of Step 2 of the algorithm that the functions instantiating members of $\mathbf{f}$ can play the role of the Skolem functions. The formula (a) of Step 2 tells us that h maps constants to themselves, and the formulas τ_P tell us, as in the proof of the opposite implication, that if $P(y_1, \ldots, y_k)$ holds in J, then $P(h(y_1), \ldots, h(y_k))$ holds in $chase_{\mathcal{M}}(I)$. So h is a homomorphism from J to $chase_{\mathcal{M}}(I)$. Therefore, $J \rightarrow chase_{\mathcal{M}}(I)$, and so $(J, I) \in \mathcal{M}^\sharp$, as desired. □

[4] In the naive chase, each time an s-t tgd γ fires, we produce new null values for the existentially-quantified variables in the conclusion of γ, no matter what the result is of other firings of s-t tgds.

COROLLARY 5.3. *There is a polynomial-time algorithm that, given a schema mapping specified by a finite set of s-t tgds, produces a strong maximum extended recovery that is specified in existential second-order logic.*

PROOF. This follows from Theorem 5.2, from the fact noted earlier, that $\mathcal{M}^\sharp$ is a strong maximum extended recovery, and from the easily-verified fact that the algorithm Max-ExtendedRecovery runs in polynomial time. □

It is an interesting question whether or not a maximum extended recovery can always be specified in the language of first-order logic. Although we have not been able to settle this question, our next theorem shows that this language must necessarily go beyond s-t tgds and some of their extensions.

THEOREM 5.4. *There is a schema mapping $\mathcal{M}$ specified by a finite set of full s-t tgds such that:*

(1) *$\mathcal{M}$ has a maximum extended recovery specified by a finite set of disjunctive tgds with inequalities.*
(2) *$\mathcal{M}$ has no maximum extended recovery specified by a set of disjunctive tgds.*
(3) *$\mathcal{M}$ has no maximum extended recovery specified by tgds with inequalities.*

PROOF. Let $\mathbf{S}$ consist of the binary relation symbol P and the unary relation symbol T, and let $\mathbf{T}$ consist of the binary relation symbol P'. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be the schema mapping such that

$$\Sigma = \{P(x, y) \to P'(x, y),\ T(x) \to P'(x, x)\}.$$

Part 1. Let $\mathcal{M}^* = (\mathbf{S}, \mathbf{T}, \Sigma^*)$ be the schema mapping, where Σ^* consists of two tgds:

$$P'(x, y) \land x \neq y \to P(x, y) \qquad P'(x, x) \to T(x) \lor P(x, x).$$

We claim that $\mathcal{M}^*$ is a maximum extended recovery of $\mathcal{M}$. To begin with, $\mathcal{M}^*$ is an extended recovery of $\mathcal{M}$ because it is easy to verify that $(I, I) \in \mathcal{M} \circ \mathcal{M}^*$, for every source instance I. By Theorem 4.14, we now need to show that $\to_{\mathcal{M}} = e(\mathcal{M}) \circ e(\mathcal{M}^*)$.

Since $\mathcal{M}^*$ is an extended recovery of $\mathcal{M}$ and since $\to_{\mathcal{M}}$ is the minimum value of the expression $e(\mathcal{M}) \circ e(\mathcal{M}')$, as $\mathcal{M}'$ varies over all extended recoveries of $\mathcal{M}$ (of which there is at least one by Theorem 4.10), we have that $\to_{\mathcal{M}} \subseteq e(\mathcal{M}) \circ e(\mathcal{M}^*)$. So, it suffices to show that $e(\mathcal{M}) \circ e(\mathcal{M}^*) \subseteq \to_{\mathcal{M}}$. In turn, it suffices to show that $\mathcal{M} \circ \to \circ \mathcal{M}^* \subseteq \to_{\mathcal{M}}$. Towards this goal, assume that $(I_1, I_2) \in \mathcal{M} \circ \to \circ \mathcal{M}^*$. We will complete the proof by showing that $chase_{\mathcal{M}}(I_1) \to chase_{\mathcal{M}}(I_2)$. Since $(I_1, I_2) \in \mathcal{M} \circ \to \circ \mathcal{M}^*$, there are target instances J_1 and J_2 such that $(I_1, J_1) \in \mathcal{M}$, $J_1 \to J_2$, and $(J_2, I_2) \in \mathcal{M}^*$. Since $chase_{\mathcal{M}}(I_1)$ is a universal solution for I_1 with respect to $\mathcal{M}$, we have that there is a homomorphism from $chase_{\mathcal{M}}(I_1)$ to J_1, hence there is a homomorphism, call it h, from $chase_{\mathcal{M}}(I_1)$ to J_2. We claim that h is also a homomorphism from $chase_{\mathcal{M}}(I_1)$ to $chase_{\mathcal{M}}(I_2)$. Let $P'(u, v)$ be a fact of $chase_{\mathcal{M}}(I_1)$. It follows that $P'(h(u), h(v))$ is a fact of J_2. If $h(u) \neq h(v)$, then $P(h(u), h(v))$ is a fact of I_2, hence $P'(h(u), h(v))$ is a fact of $chase_{\mathcal{M}}(I_2)$, which was to be shown. If $h(u) = h(v)$, then $P'(h(u), h(u))$ is a fact of J_2, hence $T(h(u))$ is a fact of I_2 or $P(h(u), h(u))$ is a fact of I_2. From this, it follows that $P'(h(u), h(u))$ (which is the same as $P'(h(u), h(v))$) is a fact of $chase_{\mathcal{M}}(I_2)$, which was to be shown. This completes the proof that $\mathcal{M}^*$ is a maximum extended recovery of $\mathcal{M}$.

Part 2. Assume now that $\mathcal{M}$ has a maximum extended recovery $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$, where Σ' is a set of disjunctive tgds. Hence, $e(\mathcal{M}) \circ e(\mathcal{M}') = \to_{\mathcal{M}}$. Let χ be a member of Σ'. We claim that the conclusion of χ must contain a disjunct whose conjuncts are of the form $\exists y T(y)$ or of the form $T(x)$. To see this, let I be the source instance with $P^I = \emptyset$ and $T^I = \{a\}$, for some constant a. Since $(I, I) \in \to_{\mathcal{M}}$ and $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$, we know that $(I, I) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. This means that there are instances I_1, J_1, J_2, I_2 such that $I \to I_1$, $(I_1, J_1) \in \mathcal{M}$, $J_1 \to J_2$, $(J_2, I_2) \in \mathcal{M}'$, and $I_2 \to I$. Since a is a constant and $T(a)$ is a fact of I, we have that $T(a)$ is a fact of I_1, hence $P'(a,a)$ is a fact of J_1, hence $P'(a,a)$ is a fact of J_2. Since $(J_2, I_2) \models \chi$, it follows that the premise of χ becomes true when all universally quantified variables in χ take value a. Consequently, there is a disjunct $\exists \mathbf{y} \varphi(\mathbf{x}, \mathbf{y})$ in the conclusion of χ, such that $\exists \mathbf{y} \varphi(\bar{\mathbf{x}}, \mathbf{y})$ holds in I_2, where $\bar{\mathbf{x}}$ assigns a to every variable in $\mathbf{x}$. Since $P^I = \emptyset$ and $I_2 \to I$, we must have that $P^{I_2} = \emptyset$, hence it must be the case that P does not occur in $\varphi(\mathbf{x}, \mathbf{y})$, which means that each conjunct of $\varphi(\mathbf{x}, \mathbf{y})$ is of the form $\exists y T(y)$ or of the form $T(x)$.

Let I_1 be $\{P(a_1, a_2), T(a_1), T(a_2)\}$ and let I_2 be $\{T(a_1), T(a_2)\}$. It is straightforward to verify that $chase_{\mathcal{M}}(I_1) = \{P'(a_1, a_2), P'(a_1, a_1), P'(a_2, a_2)\}$. and $chase_{\mathcal{M}}(I_2) = \{P'(a_1, a_1), P'(a_2, a_2)\}$. Consequently, $chase_{\mathcal{M}}(I_1) \not\to chase_{\mathcal{M}}(I_2)$, which means that $(I_1, I_2) \notin \to_{\mathcal{M}}$. We will derive a contradiction by showing that $(I_1, I_2) \in e(\mathcal{M}) \circ e(\mathcal{M}')$; in fact, we will show that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$. Clearly, $(I_1, chase_{\mathcal{M}}(I_1)) \in \mathcal{M}$. We will show that $(chase_{\mathcal{M}}(I_1), I_2) \in \mathcal{M}'$. For this, let χ be a member of Σ'. If there is an assignment of values from $chase_{\mathcal{M}}(I_1)$ to the universally quantified variables in χ so that $chase_{\mathcal{M}}(I_1)$ satisfies the premise of χ, then each variable must be assigned value a_1 or a_2. Since, by the claim in the preceding paragraph, the conclusion of χ contains a disjunct consisting of conjunctions of the form $\exists y T(y)$ or of the form $T(x)$, we have that the conclusion of χ becomes true in I_2, since I_2 contains the facts $T(a_1)$ and $T(a_2)$. This completes the proof of Part 2.

Part 3. Finally, assume that $\mathcal{M}$ has a maximum extended recovery $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$, where Σ' is a set of tgds with inequalities. So $e(\mathcal{M}) \circ e(\mathcal{M}') = \to_{\mathcal{M}}$.

Let $I_1 = \{P(a,a)\}$ and $I_2 = \{T(a)\}$ be two ground instances (a is a constant). We then have $chase_{\mathcal{M}}(I_1) = \{P'(a,a)\} = chase_{\mathcal{M}}(I_2)$, hence we have that $I_1 \to_{\mathcal{M}} I_2$ and $I_2 \to_{\mathcal{M}} I_1$; in turn, this implies that $(I_1, I_2) \in e(\mathcal{M}) \circ e(\mathcal{M}')$ and $(I_2, I_1) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. Since $(I_1, I_2) \in e(\mathcal{M}) \circ e(\mathcal{M}')$, there are I_3, J_3, J_4, and I_4 such that $I_1 \to I_3$, $(I_3, J_3) \in \mathcal{M}$, $J_3 \to J_4$, $(J_4, I_4) \in \mathcal{M}'$, and $I_4 \to I_2$. Since I_1 is a ground instance, we have that $I_1 \subseteq I_3$ and so $(I_1, J_3) \in \mathcal{M}$ (recall that $\mathcal{M}$ is specified by tgds, hence it is downward closed from the left). In turn, this implies that $chase_{\mathcal{M}}(I_1) \to J_3$ (since $chase_{\mathcal{M}}(I_1)$ is a universal solution for I_1), and so $chase_{\mathcal{M}}(I_1) \to J_4$ since $J_3 \to J_4$. Since $chase_{\mathcal{M}}(I_1) = \{P'(a,a)\}$, we have that $chase_{\mathcal{M}}(I_1) \subseteq J_4$, hence $(chase_{\mathcal{M}}(I_1), I_4) \in \mathcal{M}'$ (note that $\mathcal{M}'$ is also downward closed from the left, since $\mathcal{M}'$ is specified by tgds with inequalities). Consider now $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I_1))$; this is a single source instance (and not a set of source instances) because $\mathcal{M}'$ is specified by tgds with inequalities (no disjunctions). Since $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I_1))$ is a universal solution for $chase_{\mathcal{M}}(I_1)$ with respect to $\mathcal{M}'$, we have that $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I_1)) \to I_4$. It follows that $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I_1)) \to I_2$. In other words, we have that $chase_{\mathcal{M}'}(\{P'(a,a)\}) \to \{T(a)\}$. This implies that $chase_{\mathcal{M}'}(\{P'(a,a)\})$ does not contain any fact of the form $P(u, v)$, where u and v are constants or nulls. By reversing the roles of I_1 and I_2, an analogous argument shows that $chase_{\mathcal{M}'}(\{P'(a,a)\}) \to I_1$, which is the same as saying that $chase_{\mathcal{M}'}(\{P'(a,a)\}) \to \{P(a,a)\}$. In turn, this implies that $chase_{\mathcal{M}'}(\{P'(a,a)\})$ does not contain any fact of the form $T(u)$, where u is a constant or a null. Since $chase_{\mathcal{M}'}(\{P'(a,a)\})$ contains no fact involving P and no fact involving T, it follows that $chase_{\mathcal{M}'}(\{P'(a,a)\}) = \emptyset$. This, however, leads us to a contradiction. Indeed, if $chase_{\mathcal{M}'}(\{P'(a,a)\}) = \emptyset$, then $chase_{\mathcal{M}'}(\{P'(a,a)\}) \to P(b,b)$, where b is a constant different from a. We now have that:

—$(\{P(a,a)\}, chase_{\mathcal{M}}(\{P(a,a)\})) \in \mathcal{M}$,
—$(chase_{\mathcal{M}}(\{P(a,a)\}), chase_{\mathcal{M}'}(chase_{\mathcal{M}}(\{P(a,a)\})) \in \mathcal{M}'$, and
—$chase_{\mathcal{M}'}(chase_{\mathcal{M}}(\{P(a,a)\})) \to \{P(b,b)\}$.

It follows that $(\{P(a,a)\}, \{P(b,b)\}) \in \mathcal{M} \circ \mathcal{M}' \circ \to \subseteq \to_{\mathcal{M}}$, which is a contradiction since $\{P(a,a)\} \not\to_{\mathcal{M}} \{P(b,b)\}$. This completes the proof of the theorem. □

6. APPLICATIONS TO DATA EXCHANGE AND BEYOND

In this section, we study three applications of maximum extended recoveries: reverse data exchange, reverse query answering, and comparing schema mappings. Central to all these applications is the notion of a disjunctive relaxed chase-inverse, which generalizes the notion of a chase-inverse and provides a procedural counterpart to the notion of maximum extended recovery.

6.1. Reverse Data Exchange: Disjunctive Relaxed Chase-Inverses

We have seen that extended inverses specified by s-t tgds have an equivalent characterization as chase-inverses. This characterization shows the usefulness of extended inverses for reverse data exchange. However, for schema mappings that are not extended invertible, chase-inverses do not exist.

We shall define a relaxation of the notion of chase-inverse, called relaxed chase-inverse, which is specified by s-t tgds. More generally, we define the notion of a disjunctive relaxed chase-inverse which is specified by disjunctive s-t tgds. We argue that disjunctive relaxed chase-inverses have the desired properties for reverse data exchange even when no extended inverse exists. Furthermore, we show that maximum extended recoveries that are specified by disjunctive tgds coincide with disjunctive relaxed chase-inverses. This characterization shows, in a precise way, the benefit of maximum extended recoveries for reverse data exchange.

In the definition of a disjunctive relaxed chase-inverse and in the subsequent results, we make use of the *disjunctive chase* with disjunctive tgds. Chasing with disjunctive dependencies has been considered before in various contexts [Deutsch and Tannen 2001; Fagin et al. 2005a, 2008]. Intuitively, the disjunctive chase is an extension of the standard chase where each step generates several instances, each satisfying one of the disjuncts in the dependency that is applied. Thus, the result of the disjunctive chase is, in general, a set of instances.

Definition 6.1 (*Disjunctive Chase Step*). Let K be an instance and let σ be a disjunctive tgd:

$$\forall \mathbf{x}(\varphi(\mathbf{x}) \rightarrow \bigvee_{i=1}^{n} \exists \mathbf{y}_i \, \psi_i(\mathbf{x}, \mathbf{y}_i)).$$

We say that σ *is applicable to K with homomorphism h* if h is a homomorphism from $\varphi(\mathbf{x})$ to K such that for each $i \in \{1, \ldots, n\}$, there is no extension of h to a homomorphism h' from $\varphi(\mathbf{x}) \wedge \psi_i(\mathbf{x}, \mathbf{y}_i)$ to K. For each i with $1 \leq i \leq n$, let K_i be the union of the facts in K with the set of facts obtained by: (a) extending h to h' such that each variable in $\mathbf{y}_i$ is assigned a fresh labeled null, followed by (b) taking the image of the atoms of ψ_i under h'. We say that *the result of applying σ to K* (also called a *disjunctive chase step*) is the set $\{K_1, \ldots, K_n\}$ and write $K \xrightarrow{\sigma,h} \{K_1, \ldots, K_n\}$.

Note that in the case where σ is a tgd, the set $\{K_1, \ldots, K_n\}$ reduces to a single instance K'. We then write a chase step as $K \xrightarrow{\sigma,h} K'$.

Definition 6.2 (*Disjunctive Chase*). Let Σ be a finite set of disjunctive tgds. The *disjunctive chase of an instance K with Σ* is a tree (finite or infinite) that has K as a root and for each node K', if K' has children $K_1, \ldots, K_p$, then it must be the case that $K' \xrightarrow{\sigma,h} \{K_1, \ldots, K_p\}$ for some σ in Σ and some homomorphism h. Moreover, each leaf L in the tree has the requirement that there is no σ in Σ and no homomorphism h such that σ can be applied to L with h. When the chase tree is finite we say that *the result of the disjunctive chase of K with Σ* is the set of leaves in the chase tree.

Our case of interest is applying the disjunctive chase with Σ', where Σ' is a set of disjunctive tgds that specify a schema mapping $\mathcal{M}'$ from the target schema $\mathbf{T}$ to the source schema $\mathbf{S}$. Moreover, the input to the disjunctive chase is an instance of the form $(U, \emptyset)$, where $U = chase_{\mathcal{M}}(I)$, for some source instance I. The result of the disjunctive chase is a set $\{(U, V_1), \ldots, (U, V_m)\}$ of instances where $V_1, \ldots, V_m$ are $\mathbf{S}$-instances. (Note that the chase tree will necessarily be finite, since there is no recursion in the dependencies of $\mathcal{M}'$). If $\mathcal{V}$ denotes the set $\{V_1, \ldots, V_m\}$, we shall also say that $\mathcal{V}$ is the result of chasing U with $\mathcal{M}'$ and write $\mathcal{V} = chase_{\mathcal{M}'}(U)$.

In the proofs of this section, we shall make use of the following lemma, which is a simplified version of Lemma 6.6 of Fagin et al. [2008], shown there to hold for a slightly richer language.

LEMMA 6.3 (TRIANGLE LEMMA FOR DISJUNCTIVE TGDS). *Let σ be a disjunctive tgd and let K be an instance. Assume that σ is applicable to K with homomorphism h and let $K \xrightarrow{\sigma,h} \{K_1, \ldots, K_n\}$ be the corresponding disjunctive chase step.*

Let K^* be an instance such that (1) K^* satisfies σ, and (2) $K \rightarrow K^*$. Then there is an instance K_m in $\{K_1, \ldots, K_n\}$ such that $K_m \rightarrow K^*$.

We are now ready to define the notion of a relaxed chase-inverse. We define the more general notion of a disjunctive relaxed chase-inverse afterwards.

Definition 6.4. Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds.

(1) Let $\mathcal{M}'$ be a reverse schema mapping specified by a finite set of tgds. Then $\mathcal{M}'$ is a *relaxed chase-inverse* of $\mathcal{M}$ if for every source instance I, the following hold for the instance $V = chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$[5]:
   (a) $I \rightarrow_{\mathcal{M}} V$.
   (b) $V \rightarrow I$.

(2) Let $\mathcal{M}'$ be a reverse schema mapping specified by a finite set of disjunctive tgds. Then $\mathcal{M}'$ is a *disjunctive relaxed chase-inverse* of $\mathcal{M}$ if for every source instance I, the following hold for the set $\{V_1, \ldots, V_k\} = chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$:
   (a) We have $I \rightarrow_{\mathcal{M}} V_l$, for every $V_l \in \{V_1, \ldots, V_k\}$.
   (b) $V_i \rightarrow I$, for some $V_i \in \{V_1, \ldots, V_k\}$.

In order to interpret this definition, we first make the following parallel between $\rightarrow$ and $\rightarrow_{\mathcal{M}}$. In general, condition $I_1 \rightarrow I_2$ can be interpreted as saying that I_2 has at least as much information as I_1. The weaker condition $I_1 \rightarrow_{\mathcal{M}} I_2$ can be interpreted as saying that I_2 *exports* at least as much information as I_1. More concretely, $I_1 \rightarrow_{\mathcal{M}} I_2$ is the same as $chase_{\mathcal{M}}(I_1) \rightarrow chase_{\mathcal{M}}(I_2)$, for the case we are considering (where $\mathcal{M}$ is specified by s-t tgds). Thus, $I_1 \rightarrow_{\mathcal{M}} I_2$ says that the data that is exported via the chase with $\mathcal{M}$ from I_2 has at least as much information as the data that is exported via the chase with $\mathcal{M}$ from I_1. Then we can interpret Definition 6.4 as follows.

Part (1) of Definition 6.4 handles the case when $\mathcal{M}'$ has no disjunction and, hence, the result of reverse data exchange consists of a single source instance V. It follows from condition (b) that $V \rightarrow_{\mathcal{M}} I$. If we combine this with condition (a), we see that V exports the same information as I. Condition (b) is a soundness condition, which tells us that V has no more information than I.

Note that if we just had $\rightarrow$ instead of $\rightarrow_{\mathcal{M}}$ in condition (a), then Definition 6.4(1) would become the same as the earlier definition of a chase-inverse, since we would have homomorphisms in both directions, and hence homomorphic equivalence of V and I. Thus in effect, we relaxed the definition of a chase-inverse by replacing $\rightarrow$ with the

[5] Since in the nondisjunctive case the result of the chase is always a singleton, we write here, simply, V instead of $\{V\}$.

weaker $\rightarrow_{\mathcal{M}}$ in one of the two directions (from I to V) of homomorphic equivalence. The other direction (from V to I) cannot be weakened, since it is the soundness condition that we discussed earlier.

Generalizing to the disjunctive case, condition (a) in Definition 6.4(2) states that every instance in the set $\{V_1, \ldots, V_k\}$ that results after the reverse data exchange exports at least as much information as the original source instance I. Condition (b) states that one of the instances V_i is homomorphically contained in the original source instance I, and hence, V_i contains no extra information. Since $V_i \rightarrow I$ implies $V_i \rightarrow_{\mathcal{M}} I$, conditions (a) and (b) together imply that there is some V_i that exports exactly the same information as I. (The existence of such a V_i is what constitutes a *faithful* schema mapping, as defined in Fagin et al. [2008]. However, the definition of a disjunctive relaxed chase-inverse is stronger, since condition (b) is not necessarily implied by faithfulness.)

We illustrate the definition with a concrete example.

Example 6.5. Let us revisit the "union" schema mapping $\mathcal{M}$ of Example 3.16, where we recall that $\mathcal{M}$ is specified by the s-t tgds $P(x) \rightarrow R(x)$ and $Q(x) \rightarrow R(x)$. We have shown that $\mathcal{M}$ is not extended-invertible. This implies, by Theorem 3.22, that there can be no chase-inverse of $\mathcal{M}$. In contrast, the schema mapping $\mathcal{M}'$ specified by:

$$R(x) \rightarrow P(x) \vee Q(x),$$

is a disjunctive relaxed chase-inverse of $\mathcal{M}$. To illustrate why this is the case, consider the source instance $I = \{P(a), Q(b)\}$, where a and b are two distinct constants. Then $chase_{\mathcal{M}}(I)$ is the instance $U = \{R(a), R(b)\}$, while $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$ is the following set of four instances:

$$V_1 = \{P(a), P(b)\},\ \ V_2 = \{P(a), Q(b)\},\ \ V_3 = \{P(b), Q(a)\},\ \ V_4 = \{Q(a), Q(b)\}.$$

Let us verify that the set $\{V_1, V_2, V_3, V_4\}$ satisfies the two conditions (a) and (b) in Definition 6.4(2), with respect to the preceding source instance I. (The same type of argument can be easily extended to work for an arbitrary source instance I.)

First, it is easy to see that for each V_l, with $l = 1, \ldots, 4$, we have that $chase_{\mathcal{M}}(V_l) = U$. Since $U = chase_{\mathcal{M}}(I)$, we obtain that $chase_{\mathcal{M}}(V_l) = chase_{\mathcal{M}}(I)$, for every V_l in our set. In particular, we obtain that $I \rightarrow_{\mathcal{M}} V_l$, for every V_l in our set. Thus, condition (a) is satisfied. To verify condition (b), we just need to observe that, among the four instances in $\{V_1, V_2, V_3, V_4\}$, the instance V_2 is equal to I and, in particular, $V_2 \rightarrow I$. Intuitively, the set $\{V_1, \ldots, V_4\}$ covers all possible cases of the data that could appear in the original source instance I.

The next theorem states that, when the schema mapping $\mathcal{M}$ is extended invertible, then the relaxed chase-inverses of $\mathcal{M}$ coincide with the chase-inverses of $\mathcal{M}$. Thus, the concept of a relaxed chase-inverse (and its disjunctive generalization) is an extension of the concept of a chase-inverse.

THEOREM 6.6. *Let $\mathcal{M}$ be an extended invertible schema mapping specified by a finite set of s-t tgds, and let $\mathcal{M}'$ be a "reverse" schema mapping specified by a finite set of tgds. Then $\mathcal{M}'$ is a relaxed chase-inverse of $\mathcal{M}$ if and only if $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$.*

PROOF. Assume first that $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$. Let I be a source instance and let V be the instance $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$. Since $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$, we obtain that V is homomorphically equivalent to I. In particular, this implies both conditions (a) and (b) in Definition 6.4(2) (where condition (a) follows by the fact that $\rightarrow\ \subseteq\ \rightarrow_{\mathcal{M}}$ when $\mathcal{M}$ is specified by s-t tgds).

For the reverse direction, assume that $\mathcal{M}'$ is a relaxed chase-inverse of $\mathcal{M}$. Let I be a source instance and let V be the instance $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I))$. Since $\mathcal{M}'$ is a relaxed chase-inverse of $\mathcal{M}$, we have that $I \rightarrow_{\mathcal{M}} V$ and $V \rightarrow I$. Since $\mathcal{M}$ is extended invertible, we have that $\rightarrow_{\mathcal{M}} = \rightarrow$ (by Corollary 4.16). It follows that $I \rightarrow V$ and $V \rightarrow I$. Thus, V and I are homomorphically equivalent and, hence, $\mathcal{M}'$ is a chase-inverse of $\mathcal{M}$. □

We now point out that the preceding definition of a disjunctive relaxed chase-inverse is equivalent to (but not the same as) the definition given in the conference version [Fagin et al. 2009] of this article. Furthermore, in the conference version, we used the term *universal-faithful* for what we now call disjunctive relaxed chase-inverse. The definition in the conference version had three conditions, which turn out to be equivalent to the simpler conditions (a) and (b) in Definition 6.4(2). In particular, a third condition, called *universality*, appeared in the definition in the conference version. The next proposition, which we shall find useful later, shows that this condition is automatic.

PROPOSITION 6.7. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds and let $\mathcal{M}'$ be a disjunctive relaxed chase-inverse of $\mathcal{M}$. For every source instance I, the following holds for $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I)) = \{V_1, \ldots, V_k\}$:*

(*) For every I' such that $I \rightarrow_{\mathcal{M}} I'$, there is some $V_i \in \{V_1, \ldots, V_k\}$ such that $V_i \rightarrow I'$.

PROOF. Assume that $chase_{\mathcal{M}'}(chase_{\mathcal{M}}(I')) = \{V'_1, \ldots, V'_l\}$. Furthermore, let us denote $chase_{\mathcal{M}}(I)$ by U and $chase_{\mathcal{M}}(I')$ by U'. Since $I \rightarrow_{\mathcal{M}} I'$, we have that $U \rightarrow U'$. Assume now that V'_j is some arbitrary instance in $\{V'_1, \ldots, V'_l\}$. Since $U \rightarrow U'$, we also have that $(U, \emptyset) \rightarrow (U', V'_j)$ (where we now consider instances over the combined target and source schema). Furthermore, we have that $(U', V'_j) \in \mathcal{M}'$, since the chase, including the disjunctive one, always produces solutions. By repeatedly applying the triangle lemma for disjunctive tgds (Lemma 6.3), we obtain that there is some V_i in $\{V_1, \ldots, V_k\}$ such that $(U, V_i) \rightarrow (U', V'_j)$, and in particular, $V_i \rightarrow V'_j$. Since V'_j was arbitrarily chosen from $\{V'_1, \ldots, V'_l\}$, we proved that for every V'_j in $\{V'_1, \ldots, V'_l\}$, there is some V_i in $\{V_1, \ldots, V_k\}$ such that $V_i \rightarrow V'_j$. Since $\mathcal{M}'$ is a disjunctive relaxed chase-inverse, condition (b) must hold where we take I' to play the role of I in the definition. It follows that there is some V'_j in $\{V'_1, \ldots, V'_l\}$ such that $V'_j \rightarrow I'$. Also, we showed earlier that there is some V_i in $\{V_1, \ldots, V_k\}$ such that $V_i \rightarrow V'_j$. By composing homomorphisms, we obtain that $V_i \rightarrow I'$, for some V_i in $\{V_1, \ldots, V_k\}$. □

The next theorem shows that disjunctive relaxed chase-inverses are precisely the maximum extended recoveries that are specified by disjunctive tgds. Theorem 3.22 stated a similar relationship between the more restrictive notions of extended inverse and chase-inverse. In fact, it is not hard to see that Theorem 6.8 is a generalization of Theorem 3.22, in the sense that there is a very direct proof of Theorem 3.22 from Theorem 6.8.

THEOREM 6.8. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds and let $\mathcal{M}'$ be a "reverse" schema mapping specified by a finite set of disjunctive tgds. The following statements are equivalent:*

(1) *$\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$.*
(2) *$\mathcal{M}'$ is a disjunctive relaxed chase-inverse of $\mathcal{M}$.*

PROOF. Assume first that $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$. To prove that $\mathcal{M}'$ is a disjunctive relaxed chase-inverse of $\mathcal{M}$, let I be a source instance, let $U = chase_{\mathcal{M}}(I)$, and let $\{V_1, \ldots, V_k\} = chase_{\mathcal{M}'}(U)$. For every V_l in $\{V_1, \ldots, V_k\}$, we have that $(I, V_l) \in \mathcal{M} \circ \mathcal{M}'$, since $(I, U) \in \mathcal{M}$ and $(U, V_l) \in \mathcal{M}'$, because the chase always produces solutions. It follows that $(I, V_l) \in e(\mathcal{M}) \circ e(\mathcal{M}')$, for every V_l. This implies, by Theorem 4.14, that $(I, V_l) \in \rightarrow_{\mathcal{M}}$. Thus, we obtain $I \rightarrow_{\mathcal{M}} V_l$, for every V_l in $\{V_1, \ldots, V_k\}$, which is condition (a) in Definition 6.4(2).

We now prove condition (b) in Definition 6.4(2). Since $\mathcal{M}'$ is an extended recovery, we have that $(I, I) \in e(\mathcal{M}) \circ e(M')$. Therefore, $(I, I) \in \rightarrow \circ \mathcal{M} \circ \rightarrow \circ \mathcal{M}' \circ \rightarrow$. Thus, there exist instances I_0, J_0, J_1, and I_1 such that $I \rightarrow I_0$, $(I_0, J_0) \in \mathcal{M}$, $J_0 \rightarrow J_1$, $(J_1, I_1) \in \mathcal{M}'$, and $I_1 \rightarrow I$. We then apply the triangle lemma twice, once for tgds ($\mathcal{M}$) and once for disjunctive tgds ($\mathcal{M}'$), as follows.

First, we have that $I \rightarrow I_0$ implies $(I, \emptyset) \rightarrow (I_0, J_0)$ (where we now consider instances over the combined source and target schema). Since (I_0, J_0) satisfies all the tgds in $\mathcal{M}$, and since (I, U) is the result of chasing $(I, \emptyset)$ with $\mathcal{M}$, it follows, by applying the triangle lemma for tgds repeatedly, that $(I, U) \rightarrow (I_0, J_0)$. In particular, we obtain that $U \rightarrow J_0$.

Since also $J_0 \rightarrow J_1$, we obtain that $U \rightarrow J_1$. Now since $(J_1, I_1) \in \mathcal{M}'$, and $U \rightarrow J_1$ implies that $(U, \emptyset) \rightarrow (J_1, I_1)$, we know, by repeatedly applying the triangle lemma for disjunctive tgds, that there is some V_i in the result of the disjunctive chase of U such that $V_i \rightarrow I_1$. Furthermore, because $I_1 \rightarrow I$, it follows that $V_i \rightarrow I$. This completes the proof that (1) implies (2).

We now show that (2) implies (1). Assume that $\mathcal{M}'$ is a disjunctive relaxed chase-inverse of $\mathcal{M}$. We show that $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$, by showing that $e(\mathcal{M}) \circ e(\mathcal{M}') = \rightarrow_{\mathcal{M}}$. (Thus, we again use the characterization of maximum extended recoveries given by Theorem 4.14.) We first show that $\rightarrow_{\mathcal{M}} \subseteq e(\mathcal{M}) \circ e(\mathcal{M}')$. Assume $I \rightarrow_{\mathcal{M}} I'$. Let $\{V_1, \ldots, V_k\} = \mathit{chase}_{\mathcal{M}'}(\mathit{chase}_{\mathcal{M}}(I))$. By Proposition 6.7, condition (*) must hold. So, there is some V_i in $\{V_1, \ldots, V_k\}$ such that $V_i \rightarrow I'$. On the other hand, we have that $(I, V_i) \in \mathcal{M} \circ \mathcal{M}'$, since $(I, U) \in \mathcal{M}$ and $(U, V_i) \in \mathcal{M}'$, because the chase always produces solutions. Thus, we conclude that $(I, I') \in \mathcal{M} \circ \mathcal{M}' \circ \rightarrow$, which implies that $(I, I') \in e(\mathcal{M}) \circ e(\mathcal{M}')$.

For the opposite direction, assume that $(I, I') \in e(\mathcal{M}) \circ e(\mathcal{M}')$. Therefore, $(I, I') \in \rightarrow \circ \mathcal{M} \circ \rightarrow \circ \mathcal{M}' \circ \rightarrow$. Thus, there exist instances I_0, J_0, J_1, and I_1 such that $I \rightarrow I_0$, $(I_0, J_0) \in \mathcal{M}$, $J_0 \rightarrow J_1$, $(J_1, I_1) \in \mathcal{M}'$, and $I_1 \rightarrow I'$. Furthermore, let $U = \mathit{chase}_{\mathcal{M}}(I)$, and let $\{V_1, \ldots, V_k\} = \mathit{chase}_{\mathcal{M}'}(U)$. We then apply the triangle lemma twice (as we did earlier), once for tgds ($\mathcal{M}$) and once for disjunctive tgds ($\mathcal{M}'$), and conclude that there is some V_i in $\{V_1, \ldots, V_k\}$ such that $V_i \rightarrow I'$. At the same time, by condition (a) in Definition 6.4(2), we have that $I \rightarrow_{\mathcal{M}} V_i$. Putting the last two facts together, we obtain that $I \rightarrow_{\mathcal{M}} I'$. (We made use here of the fact that $\rightarrow_{\mathcal{M}} \circ \rightarrow = \rightarrow_{\mathcal{M}}$, which follows as in Proposition 4.12.) □

This theorem gives an additional tool for verifying whether a schema mapping $\mathcal{M}'$ is a maximum extended recovery of a schema mapping $\mathcal{M}$. For instance, the schema mapping $\mathcal{M}'$ in Example 6.5 is a maximum extended recovery for the union schema mapping $\mathcal{M}$, since $\mathcal{M}'$ is specified by a disjunctive tgd and since we have shown that $\mathcal{M}'$ is a disjunctive relaxed chase-inverse of $\mathcal{M}$.

6.2. Reverse Query Answering

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping, let q be a query over $\mathbf{T}$, and let I be an instance over $\mathbf{S}$. Since there can be more than one solution for I with respect to $\mathcal{M}$, the widely adopted semantics for answering q for I with respect to $\mathcal{M}$ is the *certain-answers* semantics [Lenzerini 2002].

Definition 6.9. The *certain-answers of q for I with respect to $\mathcal{M}$*, denoted as $\mathit{certain}_{\mathcal{M}}(q, I)$, are defined as $\bigcap_{(I,J)\in\mathcal{M}} q(J)$.

In *reverse query answering*, we assume that we have performed data exchange with $\mathcal{M}$ from a source instance I, and the query q is posed against $\mathbf{S}$ instead. This problem arises in schema evolution, for example, when old data is migrated to a new schema, but there are still queries that need to access the old data. We note that most research on query answering has focused on the direct query answering, where the query is over the target.

The reverse query answering problem is trivial if I is still available: Simply evaluate $q(I)$ to answer the query q against I. However, if I is no longer available, a natural question is whether $q(I)$ can be answered by using a maximum extended recovery. Since there are many possible source instances I' that can be returned by a maximum extended recovery $\mathcal{M}'$, a natural way of defining the semantics of reverse query answering is to consider all such possible I'. Thus, we can use $certain_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I)$ to define the semantics of reverse query answering, provided that $\mathcal{M}'$ is a maximum extended recovery of $\mathcal{M}$.

We now prove the following lemma about certain answers before our next theorem.

LEMMA 6.10. *Let $\mathcal{M}$ and $\mathcal{M}'$ be schema mappings, let q be a source query, and let I be a source instance. Then all values occurring in $certain_{e(\mathcal{M}) \circ e(M')}(q, I)$ are constants.*

PROOF. Suppose that $certain_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I)$ contains a tuple t where one of the components is a null u. Let I_1 be such that $(I, I_1) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. By the definition of certain answers, we have that $t \in q(I_1)$. Consider then the instance I_2, which is obtained from I_1 by replacing all occurrences of u by a different value v. Clearly, $I_1 \rightarrow I_2$. This fact, together with the fact that $(I, I_1) \in e(\mathcal{M}) \circ e(\mathcal{M}')$, implies that $(I, I_2) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. Therefore, t must be in $q(I_2)$. But this is not possible, since u does not occur in I_2. □

Our next theorem makes use of the notation $J_{\downarrow}$, which stands for the instance J excluding all tuples that contain at least one null. Our theorem implies that if $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$, then necessarily $certain_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I)$, where q is a conjunctive query, coincides with $q(I)_{\downarrow}$.

THEOREM 6.11. *Let $\mathcal{M}$ and $\mathcal{M}'$ be two schema mappings.*

(1) *If $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$, then $certain_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I) = q(I)_{\downarrow}$, for every source instance I and every conjunctive query q over the source schema.*
(2) *If $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$ with the property that $certain_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I) = q(I)_{\downarrow}$, for every source instance I and every conjunctive query q over the source schema, then $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$.*

PROOF. We first prove (1). Assume that $\mathcal{M}'$ is an extended inverse of $\mathcal{M}$, and let I be a source instance and q a conjunctive query. By using one containment of the equation $e(\mathcal{M}) \circ e(\mathcal{M}') = e(\text{Id})$, we have that (I, I) (which is in $e(\text{Id})$) is in $e(\mathcal{M}) \circ e(\mathcal{M}')$. Hence, $certain_{e(\mathcal{M}) \circ e(M')}(q, I) \subseteq q(I)$. Since, by Lemma 6.10, $certain_{e(\mathcal{M}) \circ e(M')}(q, I)$ contains only constants, it follows that $certain_{e(\mathcal{M}) \circ e(M')}(q, I) \subseteq q(I)_{\downarrow}$.

For the reverse inclusion, let t be a tuple in $q(I)_{\downarrow}$. (In particular, t consists entirely of constants.) Let I' be such that $(I, I') \in e(\mathcal{M}) \circ e(\mathcal{M}')$. By exploiting the other containment of the equation $e(\mathcal{M}) \circ e(\mathcal{M}') = e(\text{Id})$, we have that $(I, I') \in e(\text{Id})$ or, equivalently, $I \rightarrow I'$. Since, (1) t is in $q(I)$, (2) t consists entirely of constants, and (3) $I \rightarrow I'$, we obtain that t is also in $q(I')$. Thus, we have proved that $q(I)_{\downarrow} \subseteq certain_{e(\mathcal{M}) \circ e(M')}(q, I)$. This concludes the proof of (1).

We now prove (2). Assume that $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$, and moreover, that $certain_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I) = q(I)_{\downarrow}$, for every source instance I and every conjunctive query q over the source schema. It suffices to prove that $e(\mathcal{M}) \circ e(\mathcal{M}') \subseteq e(\text{Id})$, since the reverse inclusion $e(\text{Id}) \subseteq \text{e}(\mathcal{M}) \circ \text{e}(\mathcal{M}')$ holds by the fact that $\mathcal{M}'$ is an extended recovery of $\mathcal{M}$ (See the remark after Definition 4.3). Assume that $(I, I') \in e(\mathcal{M}) \circ e(\mathcal{M}')$, and let q^I be the canonical Boolean query of I (where nulls are replaced by existentially quantified variables). Clearly, $q^I(I) = \textbf{true}$. Furthermore, by assumption, we have that $certain_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q^I, I) = q^I(I)_{\downarrow}$. Thus, $certain_{e(\mathcal{M}) \circ e(M')}(q^I, I) = \textbf{true}$. This implies that

$q^I(I') = \mathbf{true}$, which means that $I \to I'$. We thus proved that $(I, I') \in e(\mathrm{Id})$, which concludes the proof. □

Theorem 6.11 is an indication of the goodness of the certain-answer semantics that we adopted for reverse query answering, since it shows that in the particular case of an extended invertible schema mapping $\mathcal{M}$ (where the source instance I can be recovered up to homomorphic equivalence), $\mathit{certain}_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I)$ coincides with $q(I)_{\downarrow}$ (which is the best we can do).

The next result shows that in the case when $\mathcal{M}$ is specified by s-t tgds and there exists a maximum extended recovery $\mathcal{M}'$ of $\mathcal{M}$ that is specified by disjunctive s-t tgds, we can use the chase to compute $\mathit{certain}_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I)$. In particular, assume that we are given a target instance U that is the result of the original data exchange (chase) of I with $\mathcal{M}$. We can then employ the reverse chase of U with $\mathcal{M}'$ as follows: compute the set of source instances that form the result of the reverse (disjunctive) chase, evaluate the original query over these instances, and then take the intersection of all null-free tuples. The next result shows that this gives us precisely the certain answers. We note that this result makes essential use of the fact that a maximum extended recovery specified by disjunctive tgds is a disjunctive relaxed chase-inverse.

THEOREM 6.12. *Let $\mathcal{M}$ be a schema mapping specified by a finite set of s-t tgds, let $\mathcal{M}'$ be a maximum extended recovery of $\mathcal{M}$ specified by disjunctive tgds, and let q be a conjunctive query over the source schema. Then, for every source instance I, we have that:*

$$\mathit{certain}_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I) = \left(\bigcap_{V \in \mathcal{V}} q(V) \right)_{\downarrow},$$

where $\mathcal{V} = \mathit{chase}_{\mathcal{M}'}(\mathit{chase}_{\mathcal{M}}(I))$.

PROOF. It is immediate that for each $V \in \mathcal{V}$, we have $(I, V) \in e(\mathcal{M}) \circ e(\mathcal{M}')$. Hence, $\mathit{certain}_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I) \subseteq (\bigcap_{V \in \mathcal{V}} q(V))_{\downarrow}$.

Next, we show that the reverse inclusion also holds. Let $\mathbf{c}$ be a tuple of constants in $(\bigcap_{V \in \mathcal{V}} q(V))_{\downarrow}$. Assume that $(I, I') \in e(\mathcal{M}) \circ e(\mathcal{M}')$. We shall show that $\mathbf{c} \in q(I')$. Since $\mathcal{M}'$ is a maximum extended recovery, we have, from Theorem 4.14, that $\to_{\mathcal{M}} = e(\mathcal{M}) \circ e(\mathcal{M}')$, and hence, it follows that $(I, I') \in \to_{\mathcal{M}}$. By Proposition 6.7, condition (*) must hold. In particular, we have that $V \to I'$ for some V in $\mathcal{V}$. Hence, it is also the case that $\mathbf{c} \in q(I')$, and therefore, $\mathbf{c} \in \mathit{certain}_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I)$. □

To see the theorem in action, consider the following example.

Example 6.13. Let us revisit the “union" schema mapping $\mathcal{M}$ that is specified by the s-t tgds $P(x) \to R(x)$ and $Q(x) \to R(x)$. We have shown in Example 6.5 that the schema mapping $\mathcal{M}'$ specified by the disjunctive tgd $R(x) \to P(x) \vee Q(x)$ is a disjunctive relaxed chase-inverse of $\mathcal{M}$ (and, hence, by Theorem 6.8, a maximum extended recovery of $\mathcal{M}$). Let the conjunctive query $q(x)$ over the source schema be simply $P(x)$, and consider the earlier source instance $I = \{P(a), Q(b)\}$ from Example 6.5.

To compute $\mathit{certain}_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I)$, it suffices to compute $(\bigcap_{V \in \mathcal{V}} q(V))_{\downarrow}$, where $\mathcal{V}$ is the set $\{V_1, \ldots, V_4\}$ obtained in Example 6.5 via the disjunctive chase from U, where $U = \mathit{chase}_{\mathcal{M}}(I)$. It can be seen that this intersection is empty, since there is no tuple of P that appears in every instance in $\mathcal{V}$. Hence, $\mathit{certain}_{e(\mathcal{M}) \circ e(\mathcal{M}')}(q, I) = \emptyset$. Of course, this reflects the intuition that, for schema mapping $\mathcal{M}$, there is no way to recover the original source relations, given only the result of data exchange with $\mathcal{M}$ (i.e., given U).

6.3. Comparing Schema Mappings

Definition 6.14. A schema mapping $\mathcal{M}_1$ is *less lossy* than another schema mapping $\mathcal{M}_2$ if $\rightarrow_{\mathcal{M}_1} \subseteq \rightarrow_{\mathcal{M}_2}$. We say $\mathcal{M}_1$ is *strictly less lossy* than $\mathcal{M}_2$ if $\rightarrow_{\mathcal{M}_1} \subsetneq \rightarrow_{\mathcal{M}_2}$.

If $\mathcal{M}_1$ is less lossy than $\mathcal{M}_2$, then $\mathcal{M}_1$ has a smaller information loss than $\mathcal{M}_2$ (according to Definition 4.5). Intuitively, $\mathcal{M}_1$ is more invertible than $\mathcal{M}_2$.

Example 6.15. Consider the schema mappings $\mathcal{M}_1 = (\mathbf{S}, \mathbf{T}, \Sigma_1)$ and $\mathcal{M}_2 = (\mathbf{S}, \mathbf{T}, \Sigma_2)$, where Σ_1 and Σ_2 are as follows:

$$\Sigma_1 = \{P(x, y) \rightarrow P'(x, y)\},$$
$$\Sigma_2 = \{P(x, y) \rightarrow \exists z P'(x, z),\ P(x, y) \rightarrow \exists u P'(u, y)\}.$$

The first schema mapping $\mathcal{M}_1$ copies the binary relation P to the target relation P', while the second copies each component of P separately into the same target relation P'. We now show that $\mathcal{M}_1$ is less lossy than $\mathcal{M}_2$, that is, $\rightarrow_{\mathcal{M}_1} \subseteq \rightarrow_{\mathcal{M}_2}$. Indeed, if $(I_1, I_2) \in \rightarrow_{\mathcal{M}_1}$, then it follows immediately that $I_1 \rightarrow I_2$, since $\mathcal{M}_1$ is a "copying" schema mapping. Hence, $I_1 \rightarrow_{\mathcal{M}_2} I_2$. In fact, $\mathcal{M}_1$ is a schema mapping that has no information loss (that is, $\rightarrow_{\mathcal{M}_1} = e(\mathrm{Id})$), since it is also the case that if $I_1 \rightarrow I_2$, then $(I_1, I_2) \in \rightarrow_{\mathcal{M}_1}$. Moreover, $\mathcal{M}_1$ is strictly less lossy than $\mathcal{M}_2$; let $I = \{P(1, 0)\}$ and let $I' = \{P(1, 1), P(0, 0)\}$. It is easy to see that $(I, I') \in \rightarrow_{\mathcal{M}_2}$ but $(I, I') \notin \rightarrow_{\mathcal{M}_1}$.

A theoretical framework for comparing schema mappings can be quite useful towards the justification of the design of algorithms that generate schema mappings, such as those of Fuxman et al. [2006] and Popa et al. [2002]. Each of these algorithms generates schema mappings from a visual specification of the relationship between two schemas. There are multiple ways to interpret a visual specification in general. As a simple example, the schema mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ in Example 6.15 are two possible interpretations of a visual specification that relates (via arrows) the first, and respectively, second component of P to the first, and respectively, second component of P'. We note that both schema mapping generation algorithms of Fuxman et al. [2006] and Popa et al. [2002] generate $\mathcal{M}_1$, which is the less lossy schema mapping of the two.

Finally, we characterize the property of being "less lossy," provided the schema mappings compared are specified by s-t tgds and have maximum extended recoveries specified by disjunctive tgds. As in the case of Theorem 6.12, this result makes essential use of the fact that a maximum extended recovery specified by disjunctive tgds is a disjunctive relaxed chase-inverse.

THEOREM 6.16. *Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be schema mappings specified by finite sets of s-t tgds and having the same source schema. Let $\mathcal{M}'_1$ and $\mathcal{M}'_2$ be schema mappings that are specified by disjunctive tgds and are maximum extended recoveries of $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. The following statements are equivalent:*

(1) $\rightarrow_{\mathcal{M}_1} \subseteq \rightarrow_{\mathcal{M}_2}$
(2) *For every source instance I and for every member V_1 of $chase_{\mathcal{M}'_1}(chase_{\mathcal{M}_1}(I))$, there is a member V_2 of $chase_{\mathcal{M}'_2}(chase_{\mathcal{M}_2}(I))$ such that $V_2 \rightarrow V_1$.*

PROOF. Since $\mathcal{M}'_1$ and $\mathcal{M}'_2$ are disjunctive tgds and are maximum extended recoveries of $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively, it follows from Theorem 6.8 that $\mathcal{M}'_1$ and $\mathcal{M}'_2$ are disjunctive relaxed chase-inverses of $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively.

We first show that (1) ⇒ (2). Assume that $V_1 \in \mathcal{V}_1$. By condition (a) in Definition 6.4(2), applied to $\mathcal{M}_1$ and $\mathcal{M}'_1$, we have $I \rightarrow_{\mathcal{M}_1} V_1$. Hence, we obtain $I \rightarrow_{\mathcal{M}_2} V_1$. Applying Proposition 6.7, where we let $\mathcal{M}_2$ play the role of $\mathcal{M}$ and $\mathcal{M}'_2$ play the role of $\mathcal{M}'$, we have that there exists $V_2 \in \mathcal{V}_2$ such that $V_2 \rightarrow V_1$.

We now show that (2) ⇒ (1). Suppose $I \to_{\mathcal{M}_1} I'$. Applying Proposition 6.7, this time for $\mathcal{M}_1$ and $\mathcal{M}'_1$, we have that there exists $V_1 \in \mathcal{V}_1$ such that $V_1 \to I'$. By assumption, there exists $V_2 \in \mathcal{V}_2$ such that $V_2 \to V_1$. Hence, we have $V_2 \to I'$ and therefore, it is the case that $V_2 \to_{\mathcal{M}_2} I'$. By condition (a) in Definition 6.4(2), applied to $\mathcal{M}_2$ and $\mathcal{M}'_2$, we also have that $I \to_{\mathcal{M}_2} V_2$. Therefore, we obtain $I \to_{\mathcal{M}_2} I'$, which was to be shown. □

To see the theorem in action, consider $\mathcal{M}_1$ and $\mathcal{M}_2$ of Example 6.15. The reverse schema mapping $\mathcal{M}'$ specified by $\{P'(x, y) \to P(x, y)\}$ is a maximum extended recovery for both $\mathcal{M}_1$ and $\mathcal{M}_2$. It is easy to see that for every source instance I, there is a homomorphism from $\mathit{chase}_{\mathcal{M}'}(\mathit{chase}_{\mathcal{M}_2}(I))$ to $\mathit{chase}_{\mathcal{M}'}(\mathit{chase}_{\mathcal{M}_1}(I))$. Hence, the schema mapping $\mathcal{M}_1$ is less lossy than $\mathcal{M}_2$.

Arenas et al. [2010] wrote a follow-up paper to our results here on comparing the information loss of two schema mappings. Instead of restricting attention to schema mappings specified by s-t tgds, as we do here, they allow arbitrary schema mappings. Let $\mathcal{M}_1$ and $\mathcal{M}_2$ be schema mappings. Arenas et al. say that $\mathcal{M}_1$ *transfers at least as much source information as* $\mathcal{M}_2$ if there is a schema mapping $\mathcal{N}$ such that $\mathcal{M}_2 = \mathcal{M}_1 \circ \mathcal{N}$. They show that their definition coincides to ours in the case that we consider. Concretely, they prove that if $\mathcal{M}_1$ and $\mathcal{M}_2$ are each specified by s-t tgds, then $\mathcal{M}_1$ is less lossy than $\mathcal{M}_2$ (in our sense) if and only if $\mathcal{M}_1$ transfers at least as much source information as $\mathcal{M}_2$ (in their sense). Note that Arenas et al. do not give a direct notion of information loss that applies to a single schema mapping. Instead, they give a way to compare the information loss of two schema mappings.

7. CONCLUDING REMARKS

We developed a new framework for reverse data exchange that allows source instances to contain not only constants but also nulls. In the process, we introduced and studied the notions of maximum extended recovery and information loss of a schema mapping. We believe that the results presented here may, in the long run, lead to novel applications in the design and optimization of schema mappings. An immediate problem that is left open is whether every schema mapping specified by a finite set of s-t tgds has a maximum extended recovery that is specified by a formula of first-order logic. Another interesting open problem is whether or not the main technical results concerning maximum extended recoveries of schema mappings specified by s-t tgds can be generalized to results about schema mappings with target constraints.

8. NOTATIONS

In what follows, we provide a list of notations and their corresponding meanings that were used in this article.

Notation	Meaning
$I \to I'$	There is a homomorphism from I to I', which are instances over the same schema.
$\to$	$\{(I, I') \mid I \to I'\}$
$\mathrm{Sol}_{\mathcal{M}}(I)$	$\{J \mid (I, J) \in \mathcal{M}\}$
$\mathrm{eSol}_{\mathcal{M}}(I)$	$\{J \mid (I, J) \in e(\mathcal{M})\}$
$e(\mathcal{M})$	$\to \circ \mathcal{M} \circ \to$
$I_1 \to_{\mathcal{M}} I_2$	$\mathrm{eSol}_{\mathcal{M}}(I_2) \subseteq \mathrm{eSol}_{\mathcal{M}}(I_1)$. For s-t tgds: $\mathit{chase}_{\mathcal{M}}(I_1) \to \mathit{chase}_{\mathcal{M}}(I_2)$
$\to_{\mathcal{M}}$	$\{(I_1, I_2) \mid \mathrm{eSol}_{\mathcal{M}}(I_2) \subseteq \mathrm{eSol}_{\mathcal{M}}(I_1)\}$. For s-t tgds: $\{(I_1, I_2) \mid \mathit{chase}_{\mathcal{M}}(I_1) \to \mathit{chase}_{\mathcal{M}}(I_2)\}$

REFERENCES

Abiteboul, S., Hull, R., and Vianu, V. 1995. *Foundations of Databases*. Addison-Wesley.

Afrati, F., Li, C., and Pavlaki, V. 2008. Data exchange: Query answering on incomplete data sources. In *Proceedings of the International ICST Conference on Scalable Information Systems (InfoScale)*.

Arenas, M., Pérez, J., Reutter, J. L., and Riveros, C. 2009a. Composition and inversion of schema mappings. *SIGMOD Record 38,* 3, 17–28.

Arenas, M., Pérez, J., Reutter, J. L., and Riveros, C. 2009b. Inverting schema mappings: Bridging the gap between theory and practice. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. 1018–1029.

Arenas, M., Pérez, J., Reutter, J. L., and Riveros, C. 2010. Foundations of schema mapping management. In *Proceedings of ACM Symposium on Principles of Database Systems (PODS)*. 227–238.

Arenas, M., Pérez, J., and Riveros, C. 2009. The recovery of a schema mapping: Bringing exchanged data back. *ACM Trans. Data. Syst. 34,* 4.

Beeri, C. and Vardi, M. Y. 1984. A proof procedure for data dependencies. *J. ACM 31,* 4, 718–741.

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

Bernstein, P. A., Green, T. J., Melnik, S., and Nash, A. 2008. Implementing mapping composition. *VLDB J. 17,* 2, 333–353.

Deutsch, A. and Tannen, V. 2001. Optimization properties for classes of conjunctive regular path queries. In *Proceedings of the International Workshop on Database Programming Languages (DBPL)*. 21–39.

Fagin, R. 2007. Inverting schema mappings. *ACM Trans. Data. Syst. 32,* 4.

Fagin, R., Kolaitis, P. G., Miller, R. J., and Popa, L. 2005a. Data exchange: Semantics and query answering. *Theor. Comput. Sci. 336,* 1, 89–124.

Fagin, R., Kolaitis, P. G., Popa, L., and Tan, W.-C. 2005b. Composing schema mappings: Second-order dependencies to the rescue. *ACM Trans. Data. Syst. 30,* 4, 994–1055.

Fagin, R., Kolaitis, P. G., Popa, L., and Tan, W. C. 2008. Quasi-inverses of schema mappings. *ACM Trans. Data. Syst. 33,* 2.

Fagin, R., Kolaitis, P. G., Popa, L., and Tan, W.-C. 2009. Reverse data exchange: Coping with nulls. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*. 23–32.

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

Fagin, R. and Nash, A. 2010. The structure of inverses in schema mappings. *J. ACM 57,* 6.

Fuxman, A., Hernández, M. A., Ho, C. T. H., Miller, R. J., Papotti, P., and Popa, L. 2006. Nested mappings: Schema mapping reloaded. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. 67–78.

Imieliński, T. and Lipski, Jr., W. 1983. Incomplete information and dependencies in relational databases. In *Proceedings of the ACM Symposium on Management of Data (SIGMOD)*. 178–184.

Imieliński, T. and Lipski, Jr., W. 1984. Incomplete information in relational databases. *J. ACM 31,* 4, 761–791.

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

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

Melnik, S. 2004. *Generic Model Management: Concepts and Algorithms*. Lecture Notes in Computer Science, vol. 2967, Springer.

Nash, A., Bernstein, P. A., and Melnik, S. 2005. Composition of mappings given by embedded dependencies. In *Proceedings of the ACM Symposium on Principles of Database Systems (PODS)*. 172–183.

Popa, L., Velegrakis, Y., Miller, R. J., Hernández, M. A., and Fagin, R. 2002. Translating Web data. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*. 598–609.

Received October 2009; revised January 2010, November 2010; accepted February 2011