The Recovery of a Schema Mapping: Bringing Exchanged Data Back

MARCELO ARENAS and JORGE PÉREZ
Pontificia Universidad Católica de Chile
and
CRISTIAN RIVEROS
Oxford University

A schema mapping is a specification that describes how data from a source schema is to be mapped to a target schema. Once the data has been transferred from the source to the target, a natural question is whether one can undo the process and recover the initial data, or at least part of it. In fact, it would be desirable to find a *reverse* schema mapping from target to source that specifies how to bring the exchanged data back.

In this article, we introduce the notion of a recovery of a schema mapping: it is a reverse mapping, $\mathcal{M}'$ for a mapping $\mathcal{M}$, that recovers sound data with respect to $\mathcal{M}$. We further introduce an order relation on recoveries. This allows us to choose mappings that recover the maximum amount of sound information. We call such mappings maximum recoveries. We study maximum recoveries in detail, providing a necessary and sufficient condition for their existence. In particular, we prove that maximum recoveries exist for the class of mappings specified by FO-TO-CQ source-to-target dependencies. This class subsumes the class of source-to-target tuple-generating dependencies used in previous work on data exchange. For the class of mappings specified by FO-TO-CQ dependencies, we provide an exponential-time algorithm for computing maximum recoveries, and a simplified version for full dependencies that works in quadratic time. We also characterize the language needed to express maximum recoveries, and we include a detailed comparison with the notion of inverse (and quasi-inverse) mapping previously proposed in the data exchange literature. In particular, we show that maximum recoveries strictly generalize inverses. We finally study the complexity of some decision problems related to the notions of recovery and maximum recovery.

Categories and Subject Descriptors: H.2.5 [**Database Management**]: Heterogeneous Databases—*Data translation*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Metadata management, schema mapping, data exchange, data integration, inverse, recovery, maximum recovery

M. Arenas and C. Riveros were supported by Fondecyt grants 1070732 and 1090565; J. Pérez was supported by Conicyt Ph.D. Scholarship.
Authors' addresses: M. Arenas, J. Pérez, Department of Computer Science, Pontificia Universidad Católica de Chile; email: {marenas,jperez}@ing.puc.cl; C. Riveros, Oxford University Computing Laboratory, Oxford, United Kingdom; email: cristian.riveros@comlab.ox.ac.uk.

DOI 10.1145/1620585.1620589 http://doi.acm.org/10.1145/1620585.1620589

ACM Reference Format:
Arenas, M., Perez, J., and Riveros, C. 2009. The recovery of a schema mapping: bringing exchanged data back. ACM Trans. Datab. Syst. 34, 4, Article 22 (December 2009), 48 pages.
DOI = 10.1145/1620585.1620589 http://doi.acm.org/10.1145/1620585.1620589

1. INTRODUCTION

A schema mapping is a specification that describes how data from a source schema is to be mapped to a target schema. In recent years, a lot of attention has been paid to the development of solid foundations for the problem of exchanging data using schema mappings [Fagin et al. 2005a; Libkin 2006; De Giacomo et al. 2007]. These developments are a first step towards providing a general framework for exchanging information, but they are definitely not the last one. As pointed out by Bernstein [2003], many information system problems involve not only the design and integration of complex application artifacts, but also their subsequent manipulation. This has motivated the need for the development of a general infrastructure for managing schema mappings.

A framework for managing schema mappings, called model management, was proposed by Bernstein [2003]. In this framework, operators like match, merge and compose are used to manipulate mappings [Bernstein 2003; Melnik 2004; Melnik et al. 2005]. Another important operator that naturally arises in this context is the inverse, which plays an important role in schema evolution [Bernstein and Melnik 2007]. Once the data has been transferred from the source to the target, the goal of the inverse is to recover the initial source data. If a mapping $\mathcal{M}'$ is an inverse of a mapping $\mathcal{M}$, then $\mathcal{M}'$ is an ideal mapping to bring the data exchanged through $\mathcal{M}$ back to the source.

The process of inverting schema mappings turned out to be a nontrivial task [Fagin 2007; Fagin et al. 2008]. Fagin [2007] proposed a first formal definition for what it means for a schema mapping $\mathcal{M}'$ to be an inverse of a schema mapping $\mathcal{M}$. Roughly speaking, Fagin's definition is based on the idea that a mapping composed with its inverse should be equal to the *identity schema mapping*. More formally, Fagin [2007] introduces an identity schema mapping $\overline{\text{Id}}$, suitably adapted for the case of mappings specified by source-to-target tuple-generating dependencies (st-tgds). Then he says that $\mathcal{M}'$ is an inverse of $\mathcal{M}$ if $\mathcal{M} \circ \mathcal{M}' = \overline{\text{Id}}$. This notion turns out to be rather restrictive, as it is rare, that a schema mapping possesses an inverse. In view of this limitation, in a subsequent work, Fagin et al. [2008] introduce the notion of a quasi-inverse of a schema mapping. The idea of the quasi-inverse is to relax the notion of inverse by not differentiating between source instances that are equivalent for data exchange purposes. Although numerous noninvertible schema mappings possess natural and useful quasi-inverses [Fagin et al. 2008], there are still simple mappings specified by st-tgds that have no quasi-inverse. Moreover, the notions of inverse and quasi-inverse are defined by considering identity mapping $\overline{\text{Id}}$, which is only appropriate for mappings that are closed down on the left [Fagin 2007] and, in particular, for mappings specified by st-tgds. This leaves out numerous mappings of practical interest.

In this article, we revisit the problem of inverting schema mappings. Although our motivation is similar to that of previous work, we follow a different approach. In fact, our main goal is not to define a notion of inverse mapping, but instead to give a formal definition for what it means for a schema mapping $\mathcal{M}'$ to recover *sound* information with respect to a schema mapping $\mathcal{M}$. We call such an $\mathcal{M}'$ a *recovery* of $\mathcal{M}$. We use a general definition of schema mapping, where mappings are simply defined as binary relations with pairs (I, J), where I is a source instance and J is a target instance. Our notion of recovery is applicable to this general definition of mapping. Given that, in general, there may exist many possible recoveries for a mapping, we introduce an order relation on recoveries. This naturally gives rise to the notion of maximum recovery, which is a mapping that brings back the maximum amount of sound information.

As a motivating example, consider a database with relations *Emp*(*name, works_in, lives_in*) and *DrivesWork*(*name*), the former to store names of employees and the places where they work and live, and the latter to store the names of employees who drive to work. Assume that the information about employees has to be transferred to an independent database that contains relation *Shuttle*(*name*), which stores the names of employees who take a shuttle bus to go to work. A schema mapping $\mathcal{M}_{E\text{-}S}$ between these two databases is defined by the following dependency:

$$Emp(x, y, z) \wedge y \neq z \wedge \neg DrivesWork(x) \rightarrow Shuttle(x). \tag{1}$$

An example of a reverse mapping $\mathcal{M}_1$ that recovers sound information with respect to $\mathcal{M}_{E\text{-}S}$ is $Shuttle(x) \rightarrow \exists u \exists v\, Emp(x, u, v)$; it is correct to bring back to relation *Emp* every employee in relation *Shuttle*, but since *Shuttle* does not store information about the places where employees work and live, variables u and v are existentially quantified. Furthermore, it is also correct to assume that if an employee name has been brought back from relation *Shuttle*, then the places where this employee works and lives are different. Thus, mapping $\mathcal{M}_2$ defined by $Shuttle(x) \rightarrow \exists u \exists v\, (Emp(x, u, v) \wedge u \neq v)$ is also a correct way of recovering information with respect to $\mathcal{M}_{E\text{-}S}$. On the other hand, it is clear that mapping $\mathcal{M}_3$ defined by $Shuttle(x) \rightarrow \exists u\, Emp(x, u, u)$ is not a correct way of recovering information with respect to $\mathcal{M}_{E\text{-}S}$, since $\mathcal{M}_3$ assumes that in every recovered instance, every employee in relation *Shuttle* works and lives in the same place.

Formally, an instance J is said to be a solution for an instance I under a mapping $\mathcal{M}$ if $(I, J) \in \mathcal{M}$, and the space of solutions for I under $\mathcal{M}$ is defined as the set of all instances J such that $(I, J) \in \mathcal{M}$. Then if $\mathcal{M}$ is a mapping from a source schema to a target schema and $\mathcal{M}'$ is a reverse mapping from target to source, we say that $\mathcal{M}'$ is a recovery of $\mathcal{M}$ if for every source instance I, the space of solutions for I under the composition of mappings $\mathcal{M}$ and $\mathcal{M}'$ contains I itself. That is, I must be a possible solution for itself under mapping $\mathcal{M} \circ \mathcal{M}'$. Under this definition, mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ above are recoveries of $\mathcal{M}_{E\text{-}S}$, while mapping $\mathcal{M}_3$ is not a recovery of $\mathcal{M}_{E\text{-}S}$.

Being a recovery is a sound but mild requirement. Then it would be desirable to have some criteria to compare alternative recoveries. In our motivating example, if one has to choose between $\mathcal{M}_1$ and $\mathcal{M}_2$ as a recovery of $\mathcal{M}$, then

one would probably choose $\mathcal{M}_2$, since this mapping says not only that every employee who takes a shuttle bus works and lives in some place, but also that those places must be different. Intuitively, $\mathcal{M}_2$ is *more informative than* $\mathcal{M}_1$ with respect to $\mathcal{M}$. Furthermore, if $\mathcal{M}_4$ is a mapping defined by dependency:

$$\mathit{Shuttle}(x) \rightarrow \exists u \exists v\, (\mathit{Emp}(x, u, v) \wedge u \neq v \wedge \neg \mathit{DrivesWork}(x)),$$

then $\mathcal{M}_4$ is a recovery of $\mathcal{M}_{E\text{-}S}$ that is more informative than $\mathcal{M}_2$; $\mathcal{M}_4$ additionally states that if an employee is brought back from relation *Shuttle*, then it is known that she/he does not drive to work. In general, if $\mathcal{M}'$ is a recovery of $\mathcal{M}$, then the smaller the space of solutions generated by the composition $\mathcal{M} \circ \mathcal{M}'$, the more informative $\mathcal{M}'$ is about the initial source instances. We formalize this notion by saying that $\mathcal{M}'$ is at least as informative as $\mathcal{M}''$ with respect to $\mathcal{M}$, if for every source instance I, the space of solutions for I under $\mathcal{M} \circ \mathcal{M}'$ is contained in the space of solutions for I under $\mathcal{M} \circ \mathcal{M}''$. This order on recoveries gives rise to a notion of maximum recovery. Going back to our example, it can be shown that mapping $\mathcal{M}_4$ is a maximum recovery of $\mathcal{M}_{E\text{-}S}$.

In this article, we study the notions of recovery and maximum recovery. The following are our main technical contributions:

—For the general notion of schema mapping considered in this article, we provide a necessary and sufficient condition for the existence of a maximum recovery. We use this condition to show that maximum recoveries are guaranteed to exist for a large class of schema mappings, namely for mappings specified by FO-TO-CQ source-to-target dependencies. An FO-TO-CQ dependency is a formula of the form $\forall \bar{x}(\varphi_{\mathbf{S}}(\bar{x}) \rightarrow \exists \bar{y}\, \psi_{\mathbf{T}}(\bar{x}, \bar{y}))$, where $\varphi_{\mathbf{S}}(\bar{x})$ is a first-order formula over the source schema and $\psi_{\mathbf{T}}(\bar{x}, \bar{y})$ is a conjunction of relational atoms over the target schema. Notice that every st-tgd is an FO-TO-CQ dependency. We further show that maximum recoveries exist even if we enrich the class of FO-TO-CQ dependencies with arbitrary source dependencies, equality-generating target dependencies and weakly acyclic sets of tuple-generating target dependencies.

—We provide a detailed comparison among the notions of inverse, quasi-inverse, and maximum recovery. Most notably, we show that for the class of mappings considered in Fagin [2007] and Fagin et al. [2008], if a mapping $\mathcal{M}$ is invertible, then $\mathcal{M}'$ is an inverse of $\mathcal{M}$ if and only if $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$. For this class of mappings, we also show that, if a mapping $\mathcal{M}$ is quasi-invertible, then $\mathcal{M}$ has a maximum recovery, and, furthermore, every maximum recovery of $\mathcal{M}$ is a quasi-inverse of $\mathcal{M}$.

—In this example, a maximum recovery for mapping $\mathcal{M}_{E\text{-}S}$ is obtained just by reversing the arrow of dependency (1). However in general, the process of computing maximum recoveries is more involved. For mappings specified by FO-TO-CQ dependencies, we provide an exponential-time algorithm for computing maximum recoveries. For the case of full FO-TO-CQ dependencies, that is dependencies that do not use existential quantifiers in their conclusions, we provide a quadratic-time algorithm for computing maximum recoveries. It is worth mentioning that these algorithms can also be used for computing inverses and quasi-inverses. We also investigate the language needed to

express maximum recoveries for mappings specified by FO-TO-CQ dependencies, providing justification for the dependency language used in the output of these algorithms.

—We study the complexity of some problems related to the notions of recovery and maximum recovery. We show that even for the case of st-tgds, testing whether a mapping $\mathcal{M}'$ is a recovery of a mapping $\mathcal{M}$ is undecidable. As a corollary, we obtain the same undecidability result for the notions of inverse, quasi-inverse, and maximum recovery. When restricted to full st-tgds, we prove lower complexity bounds for this problem: it is Π_2^P-complete when $\mathcal{M}$ is specified by a set of full st-tgds, and coNP-complete when both $\mathcal{M}$ and $\mathcal{M}'$ are specified by full dependencies.

Organization of the article. We start by introducing the terminology used in the article in Section 2. In Section 3, we formally define the notions of recovery and maximum recovery, and we develop several tools to study these notions. In particular, we provide in Section 3.2, a necessary and sufficient condition that characterizes the existence of maximum recoveries for general mappings. In Section 4, we use the tools developed in Section 3 to study the problem of the existence of maximum recoveries for the most common mappings used in practice. We prove positive results in Section 4.1, and some negative results in Section 4.2. In Section 5, we show how to apply the notion of maximum recovery in a significant practical situation. We compare the notion of maximum recovery with the previous notions of inverse and quasi-inverse in Section 6. In Section 7, we provide algorithms for computing maximum recoveries. In Section 8, we study the language needed to express maximum recoveries. Finally, we study in Section 9 the complexity of some decision problems related to the notions of recovery and maximum recovery. Concluding remarks are in Section 10.

This article is a substantially extended version of Arenas et al. [2008]. Besides containing the complete proofs of all the results stated in Arenas et al. [2008], this version includes new results. In Section 3.1, we provide characterizations of when a mapping $\mathcal{M}'$ is a maximum recovery of a mapping $\mathcal{M}$. All the results in this section are new. In Arenas et al. [2008], the schema evolution problem was mentioned as a natural application of the notion of maximum recovery. In this article, in Section 5, we take a step forward and formally prove that the notion of maximum recovery can be used to provide a good solution for this problem. In Section 7, we also include new and simplified versions of the algorithms for computing maximum recoveries, which exploit an interesting connection with query rewriting. In the same section, we prove a theorem about the minimum size of maximum recoveries of mappings given by st-tgds (Theorem 7.5), that was not provided in Arenas et al. [2008].

2. PRELIMINARIES

A *schema* $\mathbf{R}$ is a finite set $\{R_1, \ldots, R_k\}$ of relation symbols, with each R_i having a fixed arity n_i. Let $\mathbf{D}$ be a countably infinite domain. An instance I of $\mathbf{R}$ assigns to each relation symbol R_i of $\mathbf{R}$ a finite n_i-ary relation $R_i^I \subseteq \mathbf{D}^{n_i}$. The *domain* $\mathrm{dom}(I)$ of instance I is the set of all elements that occur in any of the relations

R_i^I. Inst($\mathbf{R}$) is defined to be the set of all instances of $\mathbf{R}$. Given instances $I, J \in$ Inst($\mathbf{R}$), we write $I \subseteq J$ to denote that, for every relation symbol R_i of $\mathbf{R}$, it holds that $R_i^I \subseteq R_i^J$.

As is customary in the data exchange literature, we consider instances with two types of values: constants and nulls Fagin et al. 2005a, 2008; Fagin 2007. More precisely, let $\mathbf{C}$ and $\mathbf{N}$ be infinite and disjoint sets of constants and nulls, respectively, and assume that $\mathbf{D} = \mathbf{C} \cup \mathbf{N}$. If we refer to a schema $\mathbf{S}$ as a *source* schema, then Inst($\mathbf{S}$) is defined to be the set of all instances of $\mathbf{S}$ that are constructed by using only elements from $\mathbf{C}$, and if we refer to a schema $\mathbf{T}$ as a *target* schema, then Inst($\mathbf{T}$) is defined as usual (instances of $\mathbf{T}$ are constructed by using elements from both $\mathbf{C}$ and $\mathbf{N}$). In this article, we use $\mathbf{S}$ to refer to a source schema and $\mathbf{T}$ to refer to a target schema.

Given schemas $\mathbf{R}_1$ and $\mathbf{R}_2$, a *schema mapping* (or just *mapping*) from $\mathbf{R}_1$ to $\mathbf{R}_2$ is a nonempty subset of Inst($\mathbf{R}_1$) $\times$ Inst($\mathbf{R}_2$). As is customary in the data exchange literature, if $\mathbf{S}$ is a source schema and $\mathbf{T}$ is a target schema, a mapping from $\mathbf{S}$ to $\mathbf{T}$ is called *source-to-target mapping* (st-mapping), and a mapping from $\mathbf{T}$ to $\mathbf{S}$ is called *target-to-source mapping* (ts-mapping) [Fagin et al. 2008].

If $\mathcal{M}$ is a schema mapping from $\mathbf{R}_1$ to $\mathbf{R}_2$ and I is an instance of $\mathbf{R}_1$, then we say that an instance J of $\mathbf{R}_2$ is a *solution for I under $\mathcal{M}$*, if $(I, J) \in \mathcal{M}$. The set of solutions for I under $\mathcal{M}$ is denoted by $\mathrm{Sol}_{\mathcal{M}}(I)$. The domain of $\mathcal{M}$, denoted by dom($\mathcal{M}$), is defined as the set of instances I such that $\mathrm{Sol}_{\mathcal{M}}(I) \neq \emptyset$. Notice that the symbol dom($\cdot$) is used to denote both the domain of an instance and of a mapping, as this does not create any confusion in the article. Furthermore, given schema mappings $\mathcal{M}_{12}$ from $\mathbf{R}_1$ to $\mathbf{R}_2$ and $\mathcal{M}_{23}$ from $\mathbf{R}_2$ to $\mathbf{R}_3$, the composition of $\mathcal{M}_{12}$ and $\mathcal{M}_{23}$ is defined as the usual composition of binary relations, that is $\mathcal{M}_{12} \circ \mathcal{M}_{23} = \{(I_1, I_3) \mid \exists I_2 : (I_1, I_2) \in \mathcal{M}_{12}$ and $(I_2, I_3) \in \mathcal{M}_{23}\}$. If $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is nonempty, then there exists a unique mapping $\mathcal{M}_{13}$, from $\mathbf{R}_1$ to $\mathbf{R}_3$, such that $\mathcal{M}_{13} = \mathcal{M}_{12} \circ \mathcal{M}_{23}$.

2.1 Dependencies and Definability of Mappings

In this article, CQ is the class of conjunctive queries and UCQ is the class of unions of conjunctive queries. If we extend these classes by allowing equalities, inequalities, or negation (of atoms), then we use superscripts $=$, $\neq$ and $\neg$, respectively. Thus, for example, $\mathrm{CQ}^{=}$ is the class of conjunctive queries with equalities and $\mathrm{UCQ}^{\neg}$ is the class of unions of conjunctive queries with negation. FO is the class of all first-order formulas with equality. Slightly abusing notation, we use $\mathbf{C}(\cdot)$ to denote a built-in unary predicate such that $\mathbf{C}(a)$ holds if and only if a is a constant, that is, $a \in \mathbf{C}$. If $\mathcal{L}$ is any of the previous query languages, then $\mathcal{L}^{\mathbf{C}}$ is the extension of $\mathcal{L}$ allowing predicate $\mathbf{C}(\cdot)$. For example, $\mathrm{CQ}^{\neq,\mathbf{C}}$ is the class of conjunctive queries with inequalities and predicate $\mathbf{C}(\cdot)$.

Dependencies. Let $\mathcal{L}_1$, $\mathcal{L}_2$ be query languages and $\mathbf{R}_1$, $\mathbf{R}_2$ be schemas with no relation symbols in common. A sentence Φ over $\mathbf{R}_1 \cup \mathbf{R}_2 \cup \{\mathbf{C}(\cdot)\}$ is an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ *dependency from* $\mathbf{R}_1$ *to* $\mathbf{R}_2$ if Φ is of the form $\forall \bar{x}\,(\varphi(\bar{x}) \rightarrow \psi(\bar{x}))$, where (1) $\bar{x}$ is the tuple of free variables in both $\varphi(\bar{x})$ and $\psi(\bar{x})$; (2) $\varphi(\bar{x})$ is an $\mathcal{L}_1$-formula over $\mathbf{R}_1 \cup \{\mathbf{C}(\cdot)\}$ if $\mathbf{C}(\cdot)$ is allowed in $\mathcal{L}_1$, and over $\mathbf{R}_1$ otherwise; and (3) $\psi(\bar{x})$ is an $\mathcal{L}_2$-formula over $\mathbf{R}_2 \cup \{\mathbf{C}(\cdot)\}$ if $\mathbf{C}(\cdot)$ is allowed in $\mathcal{L}_2$, and over $\mathbf{R}_2$ otherwise. We call

$\varphi(\bar{x})$ the *premise* of Φ, and $\psi(\bar{x})$ the *conclusion* of Φ. If $\mathbf{S}$ is a source schema and $\mathbf{T}$ is a target schema, an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependency from $\mathbf{S}$ to $\mathbf{T}$ is called an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ *source-to-target dependency* ($\mathcal{L}_1$-TO-$\mathcal{L}_2$ st-dependency), and an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependency from $\mathbf{T}$ to $\mathbf{S}$ is called an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ *target-to-source dependency* ($\mathcal{L}_1$-TO-$\mathcal{L}_2$ ts-dependency).

Three fundamental classes of dependencies for data exchange, and in particular for inverting schema mappings, are source-to-target tuple-generating dependencies (st-tgds), full source-to-target tuple-generating dependencies (full st-tgds) and target-to-source disjunctive tuple-generating dependencies with inequalities and predicate $\mathbf{C}(\cdot)$ [Fagin et al. 2005a, 2008]. The former corresponds to the class of CQ-TO-CQ st-dependencies, and the latter is an extension of the class of $\text{CQ}^{\neq,\mathbf{C}}$-TO-UCQ ts-dependencies. An FO-TO-CQ dependency is full if its conclusion does not include existential quantifiers and, thus, the class of full st-tgds corresponds to the class of full CQ-TO-CQ st-dependencies.

Semantics of dependencies, safeness. Let I be an instance of a schema $\mathbf{R} = \{R_1, \ldots, R_m\}$. Instance I can be represented as an $(\mathbf{R} \cup \{\mathbf{C}(\cdot)\})$-structure $\mathfrak{A}_I = \langle A, R_1^A, \ldots, R_m^A, \mathbf{C}^A \rangle$, where $A = \text{dom}(I)$ is the universe of $\mathfrak{A}_I$, $R_i^A = R_i^I$ for $i \in [1, m]$ and $\mathbf{C}^A = A \cap \mathbf{C}$. This representation is used to define the semantics of FO over source and target instances (here we assume familiarity with some basic notions of first-order logic).

Let $\mathbf{R}_1 = \{S_1, \ldots, S_m\}$ be a schema and I an instance of $\mathbf{R}_1$. If $\varphi(\bar{x})$ is an FO-formula over $\mathbf{R}_1 \cup \{\mathbf{C}(\cdot)\}$ and $\bar{a}$ is a tuple of elements from $\text{dom}(I)$, then we say that I satisfies $\varphi(\bar{a})$, denoted by $I \models \varphi(\bar{a})$, if and only if $\mathfrak{A}_I \models \varphi(\bar{a})$. Whenever it holds that $I \models \varphi(\bar{a})$, we say that $\bar{a}$ is an answer for φ over instance I. Furthermore, let $\mathbf{R}_2 = \{T_1, \ldots, T_n\}$ be a schema with no relation symbols in common with $\mathbf{R}_1$, and J an instance of $\mathbf{R}_2$. Then $K = (I, J)$ is an instance of $\mathbf{R}_1 \cup \mathbf{R}_2$ defined as $S_i^K = S_i^I$ and $T_j^K = T_j^J$, for $i \in [1, m]$ and $j \in [1, n]$. Notice that $\text{dom}(K) = \text{dom}(I) \cup \text{dom}(J)$. If $\varphi(\bar{x})$ is an FO-formula over $\mathbf{R}_1 \cup \mathbf{R}_2 \cup \{\mathbf{C}(\cdot)\}$ and $\bar{a}$ is a tuple of elements from $\text{dom}(I) \cup \text{dom}(J)$, then we say that (I, J) satisfies $\varphi(\bar{a})$, denoted by $(I, J) \models \varphi(\bar{a})$, if and only if $\mathfrak{A}_K \models \varphi(\bar{a})$. As usual, we say that an instance satisfies a set Σ of dependencies if the instance satisfies each dependency in Σ.

We impose the following safety condition on $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependencies. Recall that an FO-formula $\varphi(\bar{x})$ is *domain-independent* if its answer depends only on the database instance but not on the underlying domain (see Fagin [1982] for a formal definition). Let $\mathbf{R}_1$ and $\mathbf{R}_2$ be schemas with no relation symbols in common and $\Phi = \forall \bar{x}\,(\varphi(\bar{x}) \rightarrow \psi(\bar{x}))$ an $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependency from $\mathbf{R}_1$ to $\mathbf{R}_2$. Then we say that Φ is *domain-independent* if both $\varphi(\bar{x})$ and $\psi(\bar{x})$ are domain-independent. The following strategy can be used to evaluate Φ: Given instances I, J of $\mathbf{R}_1$ and $\mathbf{R}_2$, respectively, we have that $(I, J) \models \Phi$ if and only if for every tuple $\bar{a}$ of elements from $\text{dom}(I)$, if $I \models \varphi(\bar{a})$, then every component of tuple $\bar{a}$ is in $\text{dom}(J)$ and $J \models \psi(\bar{a})$. We note that this strategy cannot be used for non-domain-independent $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependencies.

Definability of mappings. Let $\mathbf{R}_1$ and $\mathbf{R}_2$ be schemas with no relation symbols in common and Σ a set of FO-sentences over $\mathbf{R}_1 \cup \mathbf{R}_2 \cup \{\mathbf{C}(\cdot)\}$. We say that

a mapping $\mathcal{M}$ from $\mathbf{R}_1$ to $\mathbf{R}_2$ is *specified* by Σ, denoted by $\mathcal{M} = (\mathbf{R}_1, \mathbf{R}_2, \Sigma)$, if for every $(I, J) \in \text{Inst}(\mathbf{R}_1) \times \text{Inst}(\mathbf{R}_2)$, we have that $(I, J) \in \mathcal{M}$ if and only if $(I, J) \models \Sigma$.

Proviso. In this article, every set Σ of dependencies is finite, and if Σ is a set of $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependencies, then we assume that every dependency in Σ is domain-independent (as defined above). Furthermore, we omit the outermost universal quantifiers from $\mathcal{L}_1$-TO-$\mathcal{L}_2$ dependencies and, thus, we write $\varphi(\bar{x}) \rightarrow \psi(\bar{x})$ instead of $\forall \bar{x}\ (\varphi(\bar{x}) \rightarrow \psi(\bar{x}))$. Finally, for the sake of readability, we write $\varphi(\bar{x}, \bar{y}) \rightarrow \psi(\bar{x})$ instead of $(\exists \bar{y}\ \varphi(\bar{x}, \bar{y})) \rightarrow \psi(\bar{x})$ in some examples, as these two formulas are equivalent.

3. RECOVERIES AND THEIR MAXIMA

Let $\mathcal{M}$ be a mapping from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$, and Id the *identity schema mapping* over $\mathbf{R}_1$, that is, $\text{Id} = \{(I, I) \mid I \in \text{Inst}(\mathbf{R}_1)\}$. When trying to invert $\mathcal{M}$, the ideal would be to find a mapping $\mathcal{M}'$ from $\mathbf{R}_2$ to $\mathbf{R}_1$ such that, $\mathcal{M} \circ \mathcal{M}' = \text{Id}$. If such a mapping exists, we know that if we use $\mathcal{M}$ to exchange data, the application of $\mathcal{M}'$ gives as a result exactly the initial source instance. Unfortunately, in most cases this ideal is impossible to reach. For example, it is impossible to obtain such an inverse if $\mathcal{M}$ is specified by a set of st-tgds [Fagin 2007]. The main problem with such an ideal definition of inverse is which, in general, no matter which $\mathcal{M}'$ we choose, we will have not one but many solutions for a source instance under $\mathcal{M} \circ \mathcal{M}'$.

If for a mapping $\mathcal{M}$, there is no mapping $\mathcal{M}_1$ such that $\mathcal{M} \circ \mathcal{M}_1 = \text{Id}$, at least we would like to find a schema mapping $\mathcal{M}_2$ that does not forbid the possibility of recovering the initial source data. That is, we would like that for every instance $I \in \text{dom}(\mathcal{M})$, the space of solutions for I under $\mathcal{M} \circ \mathcal{M}_2$ contains I itself. Such a schema mapping $\mathcal{M}_2$ is called a *recovery* of $\mathcal{M}$.

Definition 3.1. Let $\mathbf{R}_1$ and $\mathbf{R}_2$ be two schemas, $\mathcal{M}$ a mapping from $\mathbf{R}_1$ to $\mathbf{R}_2$ and $\mathcal{M}'$ a mapping from $\mathbf{R}_2$ to $\mathbf{R}_1$. Then $\mathcal{M}'$ is a *recovery* of $\mathcal{M}$ if and only if $(I, I) \in \mathcal{M} \circ \mathcal{M}'$ for every instance $I \in \text{dom}(\mathcal{M})$.

Being a recovery is a sound but mild requirement. Indeed, a schema mapping $\mathcal{M}$ from $\mathbf{R}_1$ to $\mathbf{R}_2$ always has as recoveries, for example, mappings $\mathcal{M}_1 = \text{Inst}(\mathbf{R}_2) \times \text{Inst}(\mathbf{R}_1)$ and $\mathcal{M}_2 = \mathcal{M}^{-1} = \{(J, I) \mid (I, J) \in \mathcal{M}\}$. If one has to choose between $\mathcal{M}_1$ and $\mathcal{M}_2$ as a recovery of $\mathcal{M}$, then one would probably choose $\mathcal{M}_2$ since the space of possible solutions for a source instance I under $\mathcal{M} \circ \mathcal{M}_2$ is smaller than under $\mathcal{M} \circ \mathcal{M}_1$. In fact, if there exists a mapping $\mathcal{M}_3$ such that $\mathcal{M} \circ \mathcal{M}_3 = \text{Id}$, then one would definitely prefer $\mathcal{M}_3$ over $\mathcal{M}_1$ and $\mathcal{M}_2$. In general, if $\mathcal{M}'$ is a recovery of $\mathcal{M}$, then the smaller the space of solutions generated by $\mathcal{M} \circ \mathcal{M}'$, the more informative $\mathcal{M}'$ is about the initial source instances. This notion induces an *order* among recoveries:

Definition 3.2. Let $\mathcal{M}$ be a mapping and $\mathcal{M}'$, $\mathcal{M}''$ recoveries of $\mathcal{M}$. We say that $\mathcal{M}'$ is *at least as informative as* $\mathcal{M}''$ *for* $\mathcal{M}$, and write $\mathcal{M}'' \preceq_{\mathcal{M}} \mathcal{M}'$, if and only if $\mathcal{M} \circ \mathcal{M}' \subseteq \mathcal{M} \circ \mathcal{M}''$.

Moreover, we say that $\mathcal{M}'$ and $\mathcal{M}''$ are *equally informative for* $\mathcal{M}$, denoted by $\mathcal{M}' \equiv_{\mathcal{M}} \mathcal{M}''$, if $\mathcal{M}'' \preceq_{\mathcal{M}} \mathcal{M}'$ and $\mathcal{M}' \preceq_{\mathcal{M}} \mathcal{M}''$.

Example 3.3. Let $\mathcal{M}$ be an st-mapping specified by st-tgd:

$$P(x, y) \wedge R(y, z, u) \rightarrow T(x, y, z).$$

Then the ts-mapping $\mathcal{M}_1$ specified by $T(x, y, z) \rightarrow \exists v P(x, v)$ is a recovery of $\mathcal{M}$, as well as the ts-mapping $\mathcal{M}_2$ specified by $T(x, y, z) \rightarrow P(x, y) \wedge \exists u R(y, z, u)$. Intuitively, both $\mathcal{M}_1$ and $\mathcal{M}_2$ recover sound information given the definition of $\mathcal{M}$. Furthermore, it can be shown that $\mathcal{M}_1 \preceq_{\mathcal{M}} \mathcal{M}_2$, which agrees with the intuition that $\mathcal{M}_2$ recovers more information than $\mathcal{M}_1$.

If for a mapping $\mathcal{M}$, there exists a recovery $\mathcal{M}'$ that is at least as informative as any other recovery of $\mathcal{M}$, then $\mathcal{M}'$ is the best alternative to bring exchanged data back, among all the recoveries. Intuitively, such a mapping $\mathcal{M}'$ recovers the maximum amount of sound information. Such a mapping $\mathcal{M}'$ is called a maximum recovery of $\mathcal{M}$.

Definition 3.4. Let $\mathcal{M}'$ be a recovery of a mapping $\mathcal{M}$. We say that $\mathcal{M}'$ is a *maximum recovery* of $\mathcal{M}$ if for every recovery $\mathcal{M}''$ of $\mathcal{M}$, it is the case that $\mathcal{M}'' \preceq_{\mathcal{M}} \mathcal{M}'$.

Notice that if $\mathcal{M}_1$ and $\mathcal{M}_2$ are maximum recoveries of a mapping $\mathcal{M}$, then they are equally informative for $\mathcal{M}$, that is, $\mathcal{M}_1 \equiv_{\mathcal{M}} \mathcal{M}_2$.

Example 3.5. Consider st-mapping $\mathcal{M}$ and ts-mapping $\mathcal{M}_2$ from Example 3.3. Intuitively, $\mathcal{M}_2$ is doing the best effort to recover the information exchanged by $\mathcal{M}$. In fact, it can be shown that $\mathcal{M}_2$ is a maximum recovery of $\mathcal{M}$.

3.1 Characterizing Maximum Recoveries

In this section, we focus on the problem of characterizing when a mapping $\mathcal{M}'$ is a maximum recovery of a mapping $\mathcal{M}$. For doing this, we need the notion of *reduced recovery*. A mapping $\mathcal{M}'$ is a *reduced recovery* of $\mathcal{M}$ if $\mathcal{M}'$ is a recovery of $\mathcal{M}$ and for every $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, it holds that $I_2 \in \text{dom}(\mathcal{M})$.

Example 3.6. Consider an st-mapping $\mathcal{M}$, specified by CQ-TO-CQ$^{\neq}$ st-dependency $A(x, y) \rightarrow P(x, y) \wedge x \neq y$. Notice that there are source instances that are not in the domain of $\mathcal{M}$. For example, for the instance I_1 such that $A^{I_1} = \{(1, 1), (1, 2)\}$, we have that $\text{Sol}_{\mathcal{M}}(I_1) = \emptyset$, and thus $I_1 \notin \text{dom}(\mathcal{M})$. Let $\mathcal{M}'$ be the ts-mapping specified by the ts-dependency $P(x, y) \rightarrow A(x, y)$. It is easy to see that $\mathcal{M}'$ is a recovery of $\mathcal{M}$, but not a reduced recovery of $\mathcal{M}$. In fact, the target instance J such that $P^J = \{(1, 2)\}$ has I_1 as solution. Thus, if we consider the source instance I_2 such that $A^{I_2} = \{(1, 2)\}$, we conclude that $(I_2, J) \in \mathcal{M}$ and $(J, I_1) \in \mathcal{M}'$, which implies that $(I_2, I_1) \in \mathcal{M} \circ \mathcal{M}'$ and, hence, that $\mathcal{M}'$ is not a reduced recovery of $\mathcal{M}$ since $I_1 \notin \text{dom}(\mathcal{M})$.

Consider now the ts-mapping $\mathcal{M}''$ obtained from $\mathcal{M}'$ by removing from this mapping all the tuples (J, I) such that $I \notin \text{dom}(\mathcal{M})$. Then it holds that $\mathcal{M}''$ is a reduced recovery of $\mathcal{M}$.

As shown in Example 3.6, whenever $\mathcal{M}'$ is a recovery of $\mathcal{M}$, one can extract from $\mathcal{M}'$ a reduced recovery $\mathcal{M}''$ of $\mathcal{M}$ by discarding all the pairs of instances (J, I) of $\mathcal{M}'$ such that $I \notin \text{dom}(\mathcal{M})$. The obtained reduced recovery $\mathcal{M}''$ is at least as informative as $\mathcal{M}'$ for $\mathcal{M}$ since $\mathcal{M} \circ \mathcal{M}'' \subseteq \mathcal{M} \circ \mathcal{M}'$. The following lemma formalizes this intuition. It shows that we can focus on reduced recoveries in order to find maximum recoveries.

LEMMA 3.7. *If $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$, then $\mathcal{M}'$ is a reduced recovery of $\mathcal{M}$.*

PROOF. By contradiction, assume that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$ and $\mathcal{M}'$ is not a reduced recovery of $\mathcal{M}$. Then there exists $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$ such that $I_2 \notin \text{dom}(\mathcal{M})$. Define mapping $\mathcal{M}'' \subseteq \mathcal{M}'$ as $\mathcal{M}'' = \{(J, I) \in \mathcal{M}' \mid I \in \text{dom}(\mathcal{M})\}$. Given that $\mathcal{M}'$ is a recovery of $\mathcal{M}$, we have that $\mathcal{M}''$ is a recovery of $\mathcal{M}$. Moreover, $\mathcal{M} \circ \mathcal{M}'' \subsetneq \mathcal{M} \circ \mathcal{M}'$ since $\mathcal{M}'' \subseteq \mathcal{M}'$ and $(I_1, I_2) \notin \mathcal{M} \circ \mathcal{M}''$. Thus, we have that $\mathcal{M}' \preceq_{\mathcal{M}} \mathcal{M}''$ but $\mathcal{M}'' \not\preceq_{\mathcal{M}} \mathcal{M}'$, which contradicts the fact that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$. □

From the definition of maximum recovery, one can notice that in principle, it is difficult to verify whether a mapping $\mathcal{M}'$ is a maximum recovery of a mapping $\mathcal{M}$, as it requires comparing $\mathcal{M}'$ with all the other recoveries of $\mathcal{M}$. However, the following proposition shows two alternative and useful conditions for checking whether a mapping $\mathcal{M}'$ is a maximum recovery of a mapping $\mathcal{M}$, which only depend on the structure of mappings $\mathcal{M}$ and $\mathcal{M}'$. These conditions also show that reduced recoveries are necessary for characterizing the notion of maximum recovery (notice that in (3), we are also implicitly using the notion of reduced recovery).

PROPOSITION 3.8. *Let $\mathcal{M}$ and $\mathcal{M}'$ be mappings. Then the following conditions are equivalent:*

(1) *$\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$.*
(2) *$\mathcal{M}'$ is a reduced recovery of $\mathcal{M}$ and $\mathcal{M} = \mathcal{M} \circ \mathcal{M}' \circ \mathcal{M}$.*
(3) *$\mathcal{M}'$ is a recovery of $\mathcal{M}$ and for every $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, it is the case that $\emptyset \subsetneq \text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$.*

PROOF. In this proof, we assume that $\mathcal{M}$ is a mapping from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$, and $\mathcal{M}'$ is a mapping from $\mathbf{R}_2$ to $\mathbf{R}_1$.

(1) $\Rightarrow$ (2) Assume that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$. By Lemma 3.7, we know that $\mathcal{M}'$ is a reduced recovery of $\mathcal{M}$. Given that $\mathcal{M}'$ is a recovery of $\mathcal{M}$, we have that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$ for every $I \in \text{dom}(\mathcal{M})$, which implies that $\mathcal{M} \subseteq \mathcal{M} \circ \mathcal{M}' \circ \mathcal{M}$. Thus, we only need to show that $\mathcal{M} \circ \mathcal{M}' \circ \mathcal{M} \subseteq \mathcal{M}$. On the contrary, assume that there exists $(I_1, J_1) \in \mathcal{M} \circ \mathcal{M}' \circ \mathcal{M}$ such that $(I_1, J_1) \notin \mathcal{M}$. Then, there exist instances I_2 and J_2 such that $(I_1, J_2) \in \mathcal{M}$, $(J_2, I_2) \in \mathcal{M}'$, and $(I_2, J_1) \in \mathcal{M}$. Note that $J_1 \neq J_2$ and $I_1 \neq I_2$, because we are assuming that $(I_1, J_1) \notin \mathcal{M}$. Let $\mathcal{M}^\star$ be a mapping from $\mathbf{R}_2$ to $\mathbf{R}_1$ defined as:

$$\mathcal{M}^\star = \{(J, I) \in \mathcal{M}' \mid I \neq I_2\} \cup \{(J_1, I_2)\}.$$

Given that $\mathcal{M}'$ is a recovery of $\mathcal{M}$ and $(I_2, J_1) \in \mathcal{M}$, we have that $\mathcal{M}^\star$ is a recovery of $\mathcal{M}$. Now, consider the pair (I_1, I_2). We know that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, but given that $(I_1, J_1) \notin \mathcal{M}$ and (J_1, I_2) is the only tuple in $\mathcal{M}^\star$ where I_2 appears as the second component, we have that $(I_1, I_2) \notin \mathcal{M} \circ \mathcal{M}^\star$. Thus, $\mathcal{M} \circ \mathcal{M}' \not\subseteq \mathcal{M} \circ \mathcal{M}^\star$ and, therefore, $\mathcal{M}^\star \not\preceq_{\mathcal{M}} \mathcal{M}'$. We obtain a contradiction since $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$.

(2) $\Rightarrow$ (3) Assume that $\mathcal{M}'$ is a reduced recovery of $\mathcal{M}$ and $\mathcal{M} = \mathcal{M} \circ \mathcal{M}' \circ \mathcal{M}$, and let (I_1, I_2) be in $\mathcal{M} \circ \mathcal{M}'$. Given that $\mathcal{M}'$ is a reduced recovery of $\mathcal{M}$, we have that $I_2 \in \text{dom}(\mathcal{M})$ and, therefore, $\emptyset \subsetneq \text{Sol}_{\mathcal{M}}(I_2)$. Next we show that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$.

Let $J \in \text{Sol}_{\mathcal{M}}(I_2)$. Then given that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, we have that $(I_1, J) \in \mathcal{M} \circ \mathcal{M}' \circ \mathcal{M}$. Thus, given that $\mathcal{M} = \mathcal{M} \circ \mathcal{M}' \circ \mathcal{M}$, we have that $(I_1, J) \in \mathcal{M}$ and, hence, $J \in \text{Sol}_{\mathcal{M}}(I_1)$.

(3) $\Rightarrow$ (1) Assume that $\mathcal{M}'$ is a recovery of $\mathcal{M}$ and for every $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, it is the case that $\emptyset \subsetneq \text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. For the sake of contradiction, suppose that $\mathcal{M}'$ is not a maximum recovery for $\mathcal{M}$. So there exists a recovery $\mathcal{M}''$ for $\mathcal{M}$ such that $\mathcal{M}'' \not\preceq_{\mathcal{M}} \mathcal{M}'$, that is, $\mathcal{M} \circ \mathcal{M}' \not\subseteq \mathcal{M} \circ \mathcal{M}''$. Then there exists a tuple $(I, I') \in \mathcal{M} \circ \mathcal{M}'$ such that $(I, I') \notin \mathcal{M} \circ \mathcal{M}''$. By hypothesis $\emptyset \subsetneq \text{Sol}_{\mathcal{M}}(I')$ and, thus, I' is an instance in $\text{dom}(\mathcal{M})$. Since $\mathcal{M}''$ is a recovery for $\mathcal{M}$, then we have that (I', I') is a tuple in $\mathcal{M} \circ \mathcal{M}''$. Furthermore, there exists an instance J such that $(I', J) \in \mathcal{M}$ and $(J, I') \in \mathcal{M}''$. By hypothesis, we know that $\text{Sol}_{\mathcal{M}}(I') \subseteq \text{Sol}_{\mathcal{M}}(I)$, so if $(I', J) \in \mathcal{M}$ then $(I, J) \in \mathcal{M}$. Then we have $(I, J) \in \mathcal{M}$ and $(J, I') \in \mathcal{M}''$, so we conclude that $(I, I') \in \mathcal{M} \circ \mathcal{M}''$, which is a contradiction. □

The second condition of the previous theorem is a desirable property for a reverse mapping. Intuitively, $\mathcal{M}'$ does not lose information when bringing data back from the target, if the space of solutions of every instance of the source does not change after computing $\mathcal{M} \circ \mathcal{M}'$. That is, for every instance I of $\mathbf{S}$, it holds that $\text{Sol}_{\mathcal{M}}(I) = \text{Sol}_{\mathcal{M} \circ \mathcal{M}' \circ \mathcal{M}}(I)$ (or more succinctly, $\mathcal{M} = \mathcal{M} \circ \mathcal{M}' \circ \mathcal{M}$). In general, recoveries do not satisfy this condition, but Proposition 3.8 shows that maximum recoveries satisfy it. And not only that, it also shows that maximum recoveries are the only reduced recoveries that satisfy condition $\mathcal{M} = \mathcal{M} \circ \mathcal{M}' \circ \mathcal{M}$ in the space of recoveries for $\mathcal{M}$, thus providing an alternative characterization of when $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$.

3.2 A Necessary and Sufficient Condition for the Existence of Maximum Recoveries

An important issue about the notion of recovery is whether for every mapping $\mathcal{M}$, there always exists a maximum recovery. To answer this question, we introduce the notion of *witness*, and use it to provide a necessary and sufficient condition for the existence of a maximum recovery for a mapping $\mathcal{M}$.

Definition 3.9. Let $\mathcal{M}$ be a mapping from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$ and $I \in \text{Inst}(\mathbf{R}_1)$. Then instance $J \in \text{Inst}(\mathbf{R}_2)$ is a *witness for I under $\mathcal{M}$* if for every $I' \in \text{Inst}(\mathbf{R}_1)$, if $J \in \text{Sol}_{\mathcal{M}}(I')$, then $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I')$.

Example 3.10. Consider the st-mapping $\mathcal{M}$ given by the set of st-tgds $\{A(x) \rightarrow P(x),\ B(x) \rightarrow P(x) \wedge R(x)\}$, and let I be a source instance such that

$A^I = \emptyset$ and $B^I = \{a\}$. Notice that the set of solutions for I under $\mathcal{M}$ is the set of all instances J such that $a \in P^J$ and $a \in R^J$. Consider now the target instance $J^\star$ such that $P^{J^\star} = \{a\}$. It is easy to see that if a source instance I' contains $J^\star$ as solution, then I' also has as solution every target instance J such that $a \in P^J$ and $a \in R^J$. Thus, we have that $J^\star$ is a witness for I under $\mathcal{M}$, as for every source instance I', if $J^\star \in \mathrm{Sol}_\mathcal{M}(I')$, then $\mathrm{Sol}_\mathcal{M}(I) \subseteq \mathrm{Sol}_\mathcal{M}(I')$.

Example 3.10 shows that a witness for an instance I under a mapping $\mathcal{M}$ is not necessarily a solution for I under $\mathcal{M}$. We say that J is a *witness solution* for I if J is both a witness and a solution for I. A witness solution can be considered as an *identifier* for a space of solutions; if J is a witness solution for instances I_1 and I_2, then $\mathrm{Sol}_\mathcal{M}(I_1) = \mathrm{Sol}_\mathcal{M}(I_2)$. Other identifiers for spaces of solutions have been proposed in the data exchange literature. For example, for the specific case of st-tgds, we prove in Section 4.1 that the notion of *universal solution* introduced in Fagin et al. [2005a] is stronger than the notion of witness solution, in the sense that every universal solution is a witness solution but the opposite does not hold. For other classes of st-dependencies, the notions of universal and witness solution are incomparable (see Section 4.2 for some examples).

The notion of witness together with the notion of reduced recovery can also be used to characterize when a mapping $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$. In fact, the following lemma shows that witness instances are the building blocks of maximum recoveries.

LEMMA 3.11. *Let $\mathcal{M}$ and $\mathcal{M}'$ be mappings. The following statements are equivalent:*

(1) *$\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$,*
(2) *$\mathcal{M}'$ is a reduced recovery of $\mathcal{M}$, and for every $(I_1, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$, J is a witness for I_2 under $\mathcal{M}$.*

PROOF. In this proof, we assume that $\mathcal{M}$ is a mapping from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$, and $\mathcal{M}'$ is a mapping from $\mathbf{R}_2$ to $\mathbf{R}_1$.

(1) $\Rightarrow$ (2) By Lemma 3.7, we know that $\mathcal{M}'$ is a reduced recovery of $\mathcal{M}$ and, therefore, we only need to prove that for every $(I_1, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$, J is a witness for I_2 under $\mathcal{M}$. Assume that I is an instance of $\mathbf{R}_1$ such that $J \in \mathrm{Sol}_\mathcal{M}(I)$. Since $(I, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$, we conclude that $(I, I_2) \in \mathcal{M} \circ \mathcal{M}'$, and thus, by Proposition 3.8 we obtain that $\mathrm{Sol}_\mathcal{M}(I_2) \subseteq \mathrm{Sol}_\mathcal{M}(I)$. We have shown that for every I, if $J \in \mathrm{Sol}_\mathcal{M}(I)$ then $\mathrm{Sol}_\mathcal{M}(I_2) \subseteq \mathrm{Sol}_\mathcal{M}(I)$, which proves that J is a witness of I_2.

(2) $\Rightarrow$ (1) Assume that $\mathcal{M}'$ is a reduced recovery of $\mathcal{M}$ such that, for every $(I_1, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$, J is a witness for I_2 under $\mathcal{M}$. Next we show that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$.

By Proposition 3.8 and given that $\mathcal{M}'$ is a reduced recovery of $\mathcal{M}$, we know that to prove the maximality of $\mathcal{M}'$, we only need to show that for every $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, it is the case that $\mathrm{Sol}_\mathcal{M}(I_2) \subseteq \mathrm{Sol}_\mathcal{M}(I_1)$. Take an arbitrary $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$. Thus, there exists an instance J of $\mathbf{R}_2$ such that $(I_1, J) \in \mathcal{M}$, $(J, I_2) \in \mathcal{M}'$. By hypothesis, J is a witness for I_2 under $\mathcal{M}$. By the definition of a witness

instance and given that $J \in \text{Sol}_{\mathcal{M}}(I_1)$, we conclude $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. This was to be shown. □

The previous lemma is used in the next result to provide a necessary and sufficient condition for the existence of maximum recoveries.

THEOREM 3.12. *A mapping $\mathcal{M}$ has a maximum recovery if and only if for every $I \in \text{dom}(\mathcal{M})$, there exists a witness solution for I under $\mathcal{M}$.*

PROOF. In this proof, we assume that $\mathcal{M}$ is a mapping from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$, and $\mathcal{M}'$ is a mapping from $\mathbf{R}_2$ to $\mathbf{R}_1$.

($\Rightarrow$) Assume that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$. Then by Lemma 3.11, for every $(I_1, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$, J is a witness for I_2 under $\mathcal{M}$. Let $I \in \text{dom}(\mathcal{M})$. Given that $\mathcal{M}'$ is a recovery of $\mathcal{M}$, we have that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$. Thus, there exists an instance J of $\mathbf{R}_2$ such that $(I, J) \in \mathcal{M}$, $(J, I) \in \mathcal{M}'$ and J is a witness for I under $\mathcal{M}$. We conclude that there exists $J \in \text{Sol}_{\mathcal{M}}(I)$ such that J is a witness solution for I under $\mathcal{M}$.

($\Leftarrow$) Assume that for every instance $I \in \text{dom}(\mathcal{M})$, there exists $J_I \in \text{Sol}_{\mathcal{M}}(I)$ such that J_I is a witness for I under $\mathcal{M}$, and let $\mathcal{M}^\star$ be a mapping defined as $\{(J_I, I) \mid I \in \text{dom}(\mathcal{M})\}$. It is easy to see that $\mathcal{M}^\star$ is a reduced recovery of $\mathcal{M}$. Furthermore, given that for every $(J, I) \in \mathcal{M}^\star$, J is a witness for I under $\mathcal{M}$, we conclude from Lemma 3.11 that $\mathcal{M}^\star$ is a maximum recovery of $\mathcal{M}$. □

4. ON THE EXISTENCE OF MAXIMUM RECOVERIES

In this section, we focus on source-to-target mappings, that is mappings from a source schema $\mathbf{S}$ to a target schema $\mathbf{T}$. Recall that instances of $\mathbf{S}$ are constructed by using only elements from $\mathbf{C}$ (constants), while instances of $\mathbf{T}$ are constructed by using elements from both $\mathbf{C}$ and $\mathbf{N}$ (constants and nulls). This is the most common class of mappings in the data exchange literature [Fagin et al. 2005, 2005a; Arenas et al. 2004; Afrati et al. 2008], and specifically in the literature on inverting schema mappings [Fagin 2007; Fagin et al. 2008]. We note that the recovery of an st-mapping is a target-to-source mapping.

On the positive side, we prove our main results regarding classes of st-mappings that admit maximum recoveries. Namely, we show that if $\mathcal{M}$ is an st-mapping specified by a set of FO-TO-CQ dependencies, then $\mathcal{M}$ has a maximum recovery. Furthermore, we also show that the extension of this class with source dependencies, equality-generating target dependencies, and weakly acyclic sets of tuple-generating target dependencies [Deutsch and Tannen 2003; Fagin et al. 2005a] also admits maximum recoveries (these classes of dependencies are defined in Section 4.1). These results are in sharp contrast with the results of Fagin [2007], and Fagin et al. [2008], where it was shown that even for full st-tgds, inverses and quasi-inverses are not guaranteed to exist.

On the negative side, we show that if we enrich the conclusion of FO-TO-CQ dependencies by adding inequalities, or disjunction, or negation, the existence of maximum recoveries is not guaranteed.

4.1 Positive Results

In Fagin et al. [2005a], the class of *universal solutions* for st-mappings was identified as a class of solutions that has good properties for data exchange. These solutions play an important role in this section. To formally introduce this concept, we review the necessary terminology from Fagin et al [2005a].

Let J_1 and J_2 be instances of the same schema $\mathbf{R}$. A *homomorphism* h from J_1 to J_2 is a function $h : \text{dom}(J_1) \to \text{dom}(J_2)$ such that, for every $R \in \mathbf{R}$ and every tuple $(a_1, \ldots, a_k) \in R^{J_1}$, it holds $(h(a_1), \ldots, h(a_k)) \in R^{J_2}$. Given a set $A \subseteq \mathbf{D}$, we say that a homomorphism h from J_1 to J_2 is the identity on A, if $h(a) = a$ for every $a \in A \cap \text{dom}(J_1)$. Let $\mathcal{M}$ be an st-mapping, I a source instance, and J a solution for I under $\mathcal{M}$. Then J is a *universal solution* for I under $\mathcal{M}$, if for every solution J' for I under $\mathcal{M}$, there exists a homomorphism from J to J' that is the identity on $\mathbf{C}$. The next lemma shows an important relationship between universal and witness solutions.

LEMMA 4.1.

(1) *Let $\mathcal{M}$ be an st-mapping specified by a set of* FO-TO-CQ *dependencies and I, a source instance. Then every universal solution for I under $\mathcal{M}$ is a witness solution for I under $\mathcal{M}$.*
(2) *There exists an st-mapping $\mathcal{M}$ specified by a set of st-tgds and a source instance I such that, I has a witness solution under $\mathcal{M}$ that is not a universal solution for I under $\mathcal{M}$.*

PROOF. In order to prove the first part of the lemma, let $\mathcal{M}$ be an st-mapping specified by an set of FO-TO-CQ dependencies, I a source instance and J be a universal solution for I under $\mathcal{M}$. Thus, we have that for every source instance I_1 and solution J_1 for I_1 under $\mathcal{M}$, if there exists a homomorphism from J_1 to J_2 that is the identity on I_1, then $(I_1, J_2) \in \mathcal{M}$ (this is shown in Fagin et al. [2005a] for the case of st-tgds). We use this property to prove that J is a witness solution for I under $\mathcal{M}$. Assume that $J \in \text{Sol}_{\mathcal{M}}(I')$ for an arbitrary source instance I', and let $J' \in \text{Sol}_{\mathcal{M}}(I)$. Given that J is a universal solution for I, we know that there is a homomorphism h from J to J' that is the identity on $\mathbf{C}$. Thus, we have that h is the identity on $\text{dom}(I')$, and therefore, $(I', J') \in \mathcal{M}$ since $(I', J) \in \mathcal{M}$. We conclude that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I')$. Since I' is an arbitrary source instance, we have that J is a witness for I under $\mathcal{M}$.

For the second part of the lemma, let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be an st-mapping, where $\mathbf{S} = \{P(\cdot, \cdot)\}$, $\mathbf{T} = \{R(\cdot, \cdot)\}$ and $\Sigma = \{P(x, y) \to \exists z(R(x, z) \wedge R(z, y))\}$. Assume that I is a source instance defined as $P^I = \{(a, a)\}$, where a is an arbitrary element of $\mathbf{C}$. It was shown in Fagin et al. [2005a] that, every universal solution J for I contains two tuples, (a, b) and (b, a) in R^J, with $b \in \mathbf{N}$. Thus, the solution J' for I defined as $R^{J'} = \{(a, a)\}$ is not a universal solution for I. It is not difficult to see that J' is a witness solution for I. In fact, if a source instance I' is such that $J' \in \text{Sol}_{\mathcal{M}}(I')$, then $\text{dom}(I') \subseteq \{a\}$ and, hence, I' is either the empty source instance or $I' = I$. In both cases we conclude that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I')$, which implies that J' is a witness solution for I. □

It is known that for st-mappings specified by FO-TO-CQ dependencies, universal solutions exist for every source instance [Fagin et al. 2005a; Arenas et al. 2004]. Then from Theorem 3.12 and Lemma 4.1, we obtain the following theorem.

THEOREM 4.2. *If $\mathcal{M}$ is an st-mapping specified by a set of* FO-TO-CQ *st-dependencies, then $\mathcal{M}$ has a maximum recovery.*

Example 4.3. In Fagin et al. [2008], it was shown that the schema mapping $\mathcal{M}$ specified by full st-tgd $E(x, z) \wedge E(z, y) \rightarrow F(x, y) \wedge M(z)$ has neither a quasi-inverse nor an inverse. It is possible to show that the schema mapping $\mathcal{M}'$ specified by:

$$\begin{aligned} F(x, y) &\rightarrow \exists u(E(x, u) \wedge E(u, y)), \\ M(z) &\rightarrow \exists v \exists w(E(v, z) \wedge E(z, w)), \end{aligned}$$

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

Source and target dependencies. Fix source and target schemas $\mathbf{S}$ and $\mathbf{T}$. If α is an FO-sentence over $\mathbf{S}$, then we say that α is a source FO-dependency, and if β is an FO-sentence over $\mathbf{T} \cup \{\mathbf{C}(\cdot)\}$, then we say that β is a target FO-dependency. We assume that both source and target FO-dependencies are domain-independent.

Let $\Sigma_{\mathbf{st}}, \Gamma_{\mathbf{s}}, \Gamma_{\mathbf{t}}$ be sets of source-to-target, source, and target FO-dependencies, respectively. We say that an st-mapping $\mathcal{M}$ is specified by $\Sigma_{\mathbf{st}}$, $\Gamma_{\mathbf{s}}$, and $\Gamma_{\mathbf{t}}$, and we write $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{\mathbf{st}}, \Gamma_{\mathbf{s}}, \Gamma_{\mathbf{t}})$, if $\mathcal{M}$ is specified by $\Sigma_{\mathbf{st}} \cup \Gamma_{\mathbf{s}} \cup \Gamma_{\mathbf{t}}$. Given that both $\Gamma_{\mathbf{s}}$ and $\Gamma_{\mathbf{t}}$ are sets of domain-independent sentences, we have that $(I, J) \models \Sigma_{\mathbf{st}} \cup \Gamma_{\mathbf{s}} \cup \Gamma_{\mathbf{t}}$, if and only if $(I, J) \models \Sigma_{\mathbf{st}}$, $I \models \Gamma_{\mathbf{s}}$ and $J \models \Gamma_{\mathbf{t}}$. Thus, source constraints affect the domain of an st-mapping, while target constraints affect its set of possible solutions. Notice that these roles switch when considering ts-mappings. Our next results show that maximum recoveries have good properties regarding source constraints.

LEMMA 4.4. *Let $\mathcal{M}_1$ be an st-mapping and $\mathcal{M}_1^\star$ a maximum recovery of $\mathcal{M}_1$. If $\Gamma_{\mathbf{s}}$ is a set of source* FO*-dependencies and $\mathcal{M}_2 = \{(I, J) \in \mathcal{M}_1 \mid I \models \Gamma_{\mathbf{s}}\}$, then $\mathcal{M}_2^\star = \{(J, I) \in \mathcal{M}_1^\star \mid I \models \Gamma_{\mathbf{s}}\}$ is a maximum recovery of $\mathcal{M}_2$.*

PROOF. First we show that $\mathcal{M}_2^\star$ is a recovery of $\mathcal{M}_2$. Assume that $I \in \text{dom}(\mathcal{M}_2)$. Then there exists a target instance J such that $(I, J) \in \mathcal{M}_1$ and $(I, J) \models \Gamma_{\mathbf{s}}$. Thus, we have that $I \in \text{dom}(\mathcal{M}_1)$, and then, given that $\mathcal{M}_1^\star$ is a recovery of $\mathcal{M}_1$, it holds that $(I, I) \in \mathcal{M}_1 \circ \mathcal{M}_1^\star$. Therefore, there exists a target instance K such that $(I, K) \in \mathcal{M}_1$ and $(K, I) \in \mathcal{M}_1^\star$. Thus, given that $I \models \Gamma_{\mathbf{s}}$, we obtain that $(I, I) \in \mathcal{M}_2 \circ \mathcal{M}_2^\star$. Since I is an arbitrary source instance in $\text{dom}(\mathcal{M}_2)$, we conclude that $\mathcal{M}_2^\star$ is a recovery of $\mathcal{M}_2$.

Given that $\mathcal{M}_2^\star$ is a recovery of $\mathcal{M}_2$, from Proposition 3.8 we have that $\mathcal{M}_2^\star$ is a maximum recovery of $\mathcal{M}_2$ if for every $(I_1, I_2) \in \mathcal{M}_2 \circ \mathcal{M}_2^\star$, it is the case that $\emptyset \subsetneq \text{Sol}_{\mathcal{M}_2}(I_2) \subseteq \text{Sol}_{\mathcal{M}_2}(I_1)$. Assume that $(I_1, I_2) \in \mathcal{M}_2 \circ \mathcal{M}_2^\star$. Then there exists a target instance J such that $(I_1, J) \in \mathcal{M}_2$ and $(J, I_2) \in \mathcal{M}_2^\star$. Then, we have that $(I_1, J) \in \mathcal{M}_1$ and $I_1 \models \Gamma_{\mathbf{s}}$, and that $(J, I_2) \in \mathcal{M}_1^\star$ and $I_2 \models \Gamma_{\mathbf{s}}$. Since $\mathcal{M}_1^\star$ is a maximum recovery of $\mathcal{M}_1$, we obtain from Proposition 3.8 that

$\emptyset \subsetneq \text{Sol}_{\mathcal{M}_1}(I_2) \subseteq \text{Sol}_{\mathcal{M}_1}(I_1)$. Now, notice that if an instance $I \models \Gamma_{\mathbf{s}}$, then it is straightforward to prove that $\text{Sol}_{\mathcal{M}_1}(I) = \text{Sol}_{\mathcal{M}_2}(I)$ by the construction of $\mathcal{M}_2$. Since $I_1 \models \Gamma_{\mathbf{s}}$ and $I_2 \models \Gamma_{\mathbf{s}}$, then we have that $\text{Sol}_{\mathcal{M}_1}(I_1) = \text{Sol}_{\mathcal{M}_2}(I_1)$ and $\text{Sol}_{\mathcal{M}_1}(I_2) = \text{Sol}_{\mathcal{M}_2}(I_2)$. Given that $\emptyset \subsetneq \text{Sol}_{\mathcal{M}_1}(I_2) \subseteq \text{Sol}_{\mathcal{M}_1}(I_1)$, we conclude that $\emptyset \subsetneq \text{Sol}_{\mathcal{M}_2}(I_2) \subseteq \text{Sol}_{\mathcal{M}_2}(I_1)$. This was to be shown. □

Notice that $\mathcal{M}_1$ in Lemma 4.4 is an arbitrary st-mapping. Thus, we obtain the following corollary from Theorem 4.2.

PROPOSITION 4.5. *If $\mathcal{M}$ is an st-mapping specified by a set of* FO-TO-CQ *st-dependencies together with a set of source* FO*-dependencies, then $\mathcal{M}$ has a maximum recovery.*

Example 4.6. Let $\mathcal{M}_2 = (\mathbf{S}, \mathbf{T}, \Sigma_{\mathbf{st}}, \Gamma_{\mathbf{s}})$ be an st-mapping, where $\mathbf{S} = \{A(\cdot,\cdot,\cdot)\}$, $\mathbf{T} = \{B(\cdot,\cdot), C(\cdot,\cdot)\}$ and

$$\Sigma_{\mathbf{st}} = \{A(x, y, z) \rightarrow B(x, y) \wedge C(y, z)\},$$
$$\Gamma_{\mathbf{s}} = \{A(x, y, z) \wedge A(x', y, z') \rightarrow z = z'\}.$$

Notice that $\Gamma_{\mathbf{s}}$ is a set of functional dependencies. Consider st-mapping $\mathcal{M}_1 = (\mathbf{S}, \mathbf{T}, \Sigma_{\mathbf{st}})$. Then ts-mapping specified by $\Sigma_{\mathbf{ts}} = \{B(x, y) \wedge C(y, z) \rightarrow \exists u\, A(x, y, u) \wedge \exists w\, A(w, y, z)\}$ is a maximum recovery of $\mathcal{M}_1$. Thus, we have by Lemma 4.4 that ts-mapping $\mathcal{M}_2^\star$ specified by $\Sigma_{\mathbf{ts}}$ and $\Gamma_{\mathbf{s}}$ is a maximum recovery of $\mathcal{M}_2$. We observe that $\Sigma_{\mathbf{ts}} \cup \Gamma_{\mathbf{s}}$ is logically equivalent to:

$$B(x, y) \wedge C(y, z) \rightarrow A(x, y, z), \tag{2}$$
$$A(x, y, z) \wedge A(x', y, z') \rightarrow z = z'. \tag{3}$$

In this case, we obtained what was expected; since $\Sigma_{\mathbf{st}}$ is a lossless decomposition of relation A according to $\Gamma_{\mathbf{s}}$, dependency (2) joins relations B and C to reconstruct the source instances.

We show in Section 4.2 that, if the full power of FO is allowed in target dependencies, then maximum recoveries are not guaranteed to exist. For this reason, we focus here on equality-generating dependencies and weakly acyclic tuple-generating dependencies, that are known to have good properties for data exchange. Let $\mathbf{R}$ be a schema. An *equality-generating dependency* (egd) over $\mathbf{R}$ is an FO-sentence $\forall \bar{x}(\varphi(\bar{x}) \rightarrow (x_i = x_j))$, where $\varphi(\bar{x})$ is a conjunctive query over $\mathbf{R}$, and x_i, x_j are among the variables in $\bar{x}$. A *tuple-generating dependency* (tgd) over $\mathbf{R}$ is an FO-sentence $\forall \bar{x}(\varphi(\bar{x}) \rightarrow \psi(\bar{x}))$, where both $\varphi(\bar{x})$ and $\psi(\bar{x})$ are conjunctive queries over $\mathbf{R}$.

To present the notion of weak acyclicity, we need to introduce some terminology. For a set Σ of tgds over $\mathbf{R}$, define the dependency graph G of Σ as follows:

(1) add a node (R, i) to G for every relation $R \in \mathbf{R}$ and every attribute $i \leq n_R$, where n_R is the arity of R;
(2) add an edge $(R, i) \rightarrow (T, j)$ to G if there exists a sentence $\forall \bar{x}(\varphi(\bar{x}) \rightarrow \psi(\bar{x}))$ in Σ such that if $x \in \bar{x}$ occurs in the attribute i of R in φ, then x occurs in the attribute j of T in ψ;
(3) add a special edge $(R, i) \rightarrow^* (T, j)$ to G if there exists a sentence $\forall \bar{x}(\varphi(\bar{x}) \rightarrow \psi(\bar{x}))$ in Σ such that if $x \in \bar{x}$ occurs in the attribute i of R in φ, then there

exists an existentially quantified variable y that occurs in the attribute j of T in ψ.

We say that a set Σ, of tgds is *weakly acyclic* [Deutsch and Tannen 2003; Fagin et al. 2005a] if its *dependency graph* has no cycle through a special edge. *Weak acyclicity* has been shown to be indispensable for the tractability of some important data exchange problems [Fagin et al. 2005a, 2005b; Kolaitis et al. 2006; Gottlob and Nash 2006], and thus it is a common assumption in the area.

In the following theorem, we show that maximum recoveries are guaranteed to exist in the general setting where target egds and weakly acyclic sets of target tgds are allowed.

THEOREM 4.7. *Let* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{\mathbf{st}}, \Gamma_{\mathbf{s}}, \Gamma_{\mathbf{t}})$ *be an st-mapping, where* $\Sigma_{\mathbf{st}}$ *is a set of* FO-TO-CQ *st-dependencies,* $\Gamma_{\mathbf{s}}$ *is a set of source* FO*-dependencies, and* $\Gamma_{\mathbf{t}}$ *is the union of a set of target egds and a weakly acyclic set of target tgds. Then* $\mathcal{M}$ *has a maximum recovery.*

PROOF. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{\mathbf{st}}, \Gamma_{\mathbf{s}}, \Gamma_{\mathbf{t}})$ be an st-mapping, where $\Sigma_{\mathbf{st}}$ is a set of FO-TO-CQ dependencies, $\Gamma_{\mathbf{s}}$ is a set of source dependencies, and $\Gamma_{\mathbf{t}}$ is the union of a set of egds and a weakly acyclic set of tgds, and let $\mathcal{M}_1 = (\mathbf{S}, \mathbf{T}, \Sigma_{\mathbf{st}}, \Gamma_{\mathbf{t}})$. From Lemma 4.4, we know that in order to prove that $\mathcal{M}$ has a maximum recovery, it is enough to show that $\mathcal{M}_1$ has a maximum recovery. Next we prove the latter.

Given that $\Sigma_{\mathbf{st}}$ is a set of FO-TO-CQ dependencies and $\Gamma_{\mathbf{t}}$ is the the union of a set of egds and a weakly acyclic set of tgds, for every instance $I \in \text{dom}(\mathcal{M}_1)$, we have that (1) there exists a universal solution J for I under $\mathcal{M}_1$, and (2) for every solution J_1 for I under $\mathcal{M}$ and instance J_2 of $\mathbf{T}$ that satisfies $\Gamma_{\mathbf{t}}$, if there exists a homomorphism from J_1 to J_2 that is the identity on I, then $(I, J_2) \in \mathcal{M}$ (these two properties are proved in Fagin et al. [2005a] for the case of st-tgds). We use these conditions to prove that $\mathcal{M}_1$ has a maximum recovery.

From Theorem 3.12, we know that to prove $\mathcal{M}_1$ has a maximum recovery, it is enough to show that every $I_1 \in \text{dom}(\mathcal{M}_1)$ has a witness solution under $\mathcal{M}_1$. Let I_1 be an instance of $\mathbf{S}$ such that $I_1 \in \text{dom}(\mathcal{M}_1)$ and J_1 a universal solution for I_1. Next we show that J_1 is a witness solution for I_1. Assume that $J_1 \in \text{Sol}_{\mathcal{M}_1}(I_2)$ for an arbitrary source instance I_2. We need to prove that $\text{Sol}_{\mathcal{M}_1}(I_1) \subseteq \text{Sol}_{\mathcal{M}_1}(I_2)$. Let $J \in \text{Sol}_{\mathcal{M}_1}(I_1)$. Given that J_1 is a universal solution for I_1, we know that there is a homomorphism h from J_1 to J that is the identity on $\mathbf{C}$. Furthermore, given that $J \in \text{Sol}_{\mathcal{M}_1}(I_1)$, we have that $J \models \Gamma_{\mathbf{t}}$. Thus, we conclude that $J \in \text{Sol}_{\mathcal{M}_1}(I_2)$, since $(I_2, J_1) \in \mathcal{M}_1$, $J \models \Gamma_{\mathbf{t}}$ and h is a homomorphism from J_1 to J that is the identity on $\text{dom}(I_2)$ (given that h is the identity on $\mathbf{C}$ and $\text{dom}(I_2) \subseteq \mathbf{C}$). This concludes the proof of the theorem. □

Notice that the positive results of this section do not say anything about the language needed to express maximum recoveries. In Sections 7 and 8, we study this problem.

4.2 Negative Results

In Section 4.1, we prove that FO-TO-CQ st-mappings have maximum recoveries using the relationship between universal and witness solutions shown in Lemma 4.1. If we go beyond CQ in the conclusions of dependencies, these notions become incomparable. For example, consider an st-mapping $\mathcal{M}_1$ specified by CQ-TO-CQ$^{\neq}$ dependencies $P(x) \rightarrow \exists y\, R(x, y)$ and $S(x) \rightarrow \exists y\, (R(x, y) \wedge x \neq y)$, and let I be a source instance such that $P^I = \{a\}$. Target instance J_1 such that $R^{J_1} = \{(a, n)\}$, with $n \in \mathbf{N}$, is a universal solution but not a witness for I, while target instance J_2 such that $R^{J_2} = \{(a, a)\}$ is a witness but not a universal solution for I. In this example, every source instance has a witness solution, and thus $\mathcal{M}_1$ has a maximum recovery. In fact, dependencies $R(x, y) \rightarrow P(x) \vee S(x)$ and $R(x, y) \wedge x \neq y \rightarrow S(x)$ specify a maximum recovery of $\mathcal{M}_1$. As a second example, consider st-mapping $\mathcal{M}_2$ specified by CQ-TO-UCQ dependency $P(x) \rightarrow R(x) \vee S(x)$. In this case, every source instance has a witness solution, and only the empty source instance has a universal solution. In fact, dependencies $R(x) \rightarrow P(x)$ and $S(x) \rightarrow P(x)$ specify a maximum recovery of $\mathcal{M}_2$.

We have shown examples of mappings that have maximum recoveries and are specified by dependencies with inequalities and disjunctions in the conclusions. However, the following proposition shows that this is not a general phenomenon. If we slightly enrich the language used in the conclusions of FO-TO-CQ dependencies, then the existence of maximum recoveries is not guaranteed, even if premises are restricted to be conjunctive queries.

PROPOSITION 4.8. *There exist st-mappings specified by* (1) CQ-TO-CQ$^{\neq}$, (2) CQ-TO-UCQ, *and* (3) CQ-TO-CQ$^{\neg}$ *dependencies, that have no maximum recoveries.*

PROOF. (1) CQ-TO-CQ$^{\neq}$: Let $\mathbf{S} = \{F(\cdot), G(\cdot), H(\cdot)\}$, $\mathbf{T} = \{R(\cdot, \cdot)\}$, and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ an st-mapping specified by the following set Σ of CQ-TO-CQ$^{\neq}$ st-dependencies:

$$\begin{aligned} F(x) &\rightarrow R(x, x), \\ G(x) &\rightarrow \exists y R(x, y), \\ H(x) &\rightarrow \exists y (R(x, y) \wedge x \neq y). \end{aligned}$$

Let I_1 be an instance of $\mathbf{S}$ such that $G^{I_1} = \{a\}$ and $F^{I_1} = H^{I_1} = \emptyset$, where a is an arbitrary element of $\mathbf{C}$. Next we show that there is no $J \in \mathrm{Sol}_{\mathcal{M}}(I_1)$ such that J is a witness for I_1 under $\mathcal{M}$. Then by Theorem 3.12, $\mathcal{M}$ has no maximum recovery. For the sake of contradiction, assume that J is a witness solution for I_1 under $\mathcal{M}$. Given that J is a solution for I_1, we have that $(a, b) \in R^J$ for some b. We need to consider two cases, depending on b. If $b = a$, then the instance I_2 where $F^{I_2} = \{a\}$ and $G^{I_2} = H^{I_2} = \emptyset$ is such that $J \in \mathrm{Sol}_{\mathcal{M}}(I_2)$, but it is not the case that $\mathrm{Sol}_{\mathcal{M}}(I_1) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_2)$. If $b \neq a$, then the instance I_3 where $H^{I_3} = \{a\}$ and $F^{I_3} = G^{I_3} = \emptyset$ is such that $J \in \mathrm{Sol}_{\mathcal{M}}(I_3)$, but it is not the case that $\mathrm{Sol}_{\mathcal{M}}(I_1) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_3)$.

(2) CQ-TO-UCQ: Let $\mathbf{S} = \{F(\cdot), G(\cdot), H(\cdot)\}$, $\mathbf{T} = \{R(\cdot), S(\cdot), T(\cdot)\}$ and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ an st-mapping specified by the following set Σ of CQ-TO-UCQ

st-dependencies:

$$\begin{aligned} F(x) &\rightarrow R(x) \vee S(x), \\ G(x) &\rightarrow S(x) \vee T(x), \\ H(x) &\rightarrow T(x) \vee R(x). \end{aligned}$$

Let I be an instance of $\mathbf{S}$ such that $F^I = \{a\}$ and $G^I = H^I = \emptyset$, where a is an arbitrary element of $\mathbf{C}$. Next we show that there is no $J \in \mathrm{Sol}_{\mathcal{M}}(I)$ such that J is a witness for I under $\mathcal{M}$. For the sake of contradiction, assume that J is a witness solution for I under $\mathcal{M}$. Given that J is a solution for I, we have that $a \in R^J$ or $a \in S^J$. Assume without loss of generality that $a \in S^J$, and consider instance I_1 of $\mathbf{S}$ such that $G^{I_1} = \{a\}$ and $F^{I_1} = H^{I_1} = \emptyset$. Then we have that $J \in \mathrm{Sol}_{\mathcal{M}}(I_1)$ and, therefore, $\mathrm{Sol}_{\mathcal{M}}(I) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$ since J is a witness solution for I under $\mathcal{M}$. Let J_1 be an instance of $\mathbf{T}$ such that $R^{J_1} = \{a\}$ and $S^{J_1} = T^{J_1} = \emptyset$. We have that $J_1 \in \mathrm{Sol}_{\mathcal{M}}(I)$ but $J_1 \notin \mathrm{Sol}_{\mathcal{M}}(I_1)$, which contradicts the fact that $\mathrm{Sol}_{\mathcal{M}}(I) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$.

Given that there is no witness solution for I under $\mathcal{M}$, we conclude by Theorem 3.12 that $\mathcal{M}$ does not have a maximum recovery.

(3) CQ-TO-CQ$^{\neg}$: Let $\mathbf{S} = \{F(\cdot), G(\cdot), H(\cdot)\}$, $\mathbf{T} = \{R(\cdot), S(\cdot)\}$, and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ an st-mapping specified by the following set Σ of CQ-TO-CQ$^{\neg}$ st-dependencies:

$$\begin{aligned} F(x) &\rightarrow R(x), \\ G(x) &\rightarrow R(x) \wedge S(x), \\ H(x) &\rightarrow R(x) \wedge \neg S(x). \end{aligned}$$

Let I_1 be an instance of $\mathbf{S}$ such that $F^{I_1} = \{a\}$ and $G^{I_1} = H^{I_1} = \emptyset$, where a is an arbitrary element of $\mathbf{C}$. Next we show that there is no $J \in \mathrm{Sol}_{\mathcal{M}}(I_1)$ such that J is a witness for I_1 under $\mathcal{M}$. For the sake of contradiction, assume that J is a witness solution for I_1 under $\mathcal{M}$. Given that J is a solution for I_1, we have that $a \in R^J$. We need to consider two cases depending on whether a belongs to S^J or not. If $a \in S^J$, then the instance I_2 where $G^{I_2} = \{a\}$ and $H^{I_2} = F^{I_2} = \emptyset$ is such that $J \in \mathrm{Sol}_{\mathcal{M}}(I_2)$, but it is not the case that $\mathrm{Sol}_{\mathcal{M}}(I_1) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_2)$. If $a \notin S^J$, then the instance I_3 where $H^{I_3} = \{a\}$ and $F^{I_3} = G^{I_3} = \emptyset$ is such that $J \in \mathrm{Sol}_{\mathcal{M}}(I_3)$, but it is not the case that $\mathrm{Sol}_{\mathcal{M}}(I_1) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_3)$.

Given that there is no witness solution for I_1 under $\mathcal{M}$, we conclude by Theorem 3.12 that $\mathcal{M}$ does not have a maximum recovery. □

We conclude this section by showing that, if the full power of FO is allowed in target dependencies, then maximum recoveries are not guaranteed to exist.

PROPOSITION 4.9. *There exists an st-mapping specified by a set of st-tgds plus a set of target* FO*-dependencies that has no maximum recovery.*

PROOF. Let $\mathbf{S} = \{F(\cdot), G(\cdot), H(\cdot)\}$, $\mathbf{T} = \{R(\cdot), S(\cdot), T(\cdot)\}$ and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{\mathbf{st}}, \Gamma_{\mathbf{t}})$, an st-mapping specified by the following set $\Sigma_{\mathbf{st}}$ of st-tgds:

$$\begin{aligned} F(x) &\rightarrow R(x), \\ G(x) &\rightarrow S(x), \\ H(x) &\rightarrow T(x), \end{aligned}$$

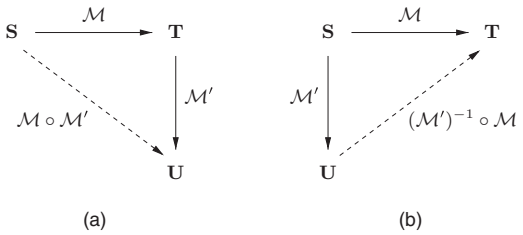

Fig. 1. The schema evolution problem.

and the following set $\Gamma_{\mathbf{t}}$ of target FO-dependencies:

$$R(x) \rightarrow S(x) \vee T(x).$$

Let I be an instance of $\mathbf{S}$ such that $F^I = \{a\}$ and $G^I = H^I = \emptyset$, where a is an arbitrary element of $\mathbf{C}$. Next we show that there is no $J \in \mathrm{Sol}_{\mathcal{M}}(I)$ such that J is a witness for I under $\mathcal{M}$. For the sake of contradiction, assume that J is a witness solution for I under $\mathcal{M}$. Given that J is a solution for I, we have that $a \in S^J$ or $a \in T^J$ (since $a \in R^J$). Assume without loss of generality that $a \in S^J$, and consider instance I_1, of $\mathbf{S}$ such that $G^{I_1} = \{a\}$ and $F^{I_1} = H^{I_1} = \emptyset$. Then we have that $J \in \mathrm{Sol}_{\mathcal{M}}(I_1)$, and therefore $\mathrm{Sol}_{\mathcal{M}}(I) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$, since J is a witness solution for I under $\mathcal{M}$. Let J_1 be an instance of $\mathbf{T}$ such that $R^{J_1} = T^{J_1} = \{a\}$ and $S^{J_1} = \emptyset$. We have that $J_1 \in \mathrm{Sol}_{\mathcal{M}}(I)$ but $J_1 \notin \mathrm{Sol}_{\mathcal{M}}(I_1)$, which contradicts the fact that $\mathrm{Sol}_{\mathcal{M}}(I) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$.

Given that there is no witness solution for I under $\mathcal{M}$, we conclude by Theorem 3.12, that $\mathcal{M}$ does not have a maximum recovery. □

5. AN APPLICATION OF MAXIMUM RECOVERIES

One of the main reasons for the study of the issues of composing and inverting schema mappings is to solve the schema evolution problem [Fagin et al. 2005; Fagin 2007]. Two main scenarios have been identified for this problem, which are shown in Figure 1. In scenario (a), a mapping $\mathcal{M}$ from a schema $\mathbf{S}$ to a schema $\mathbf{T}$ has already been constructed, and it has been decided that target schema $\mathbf{T}$ will be replaced by a new schema $\mathbf{U}$. In particular, the relationship between schemas $\mathbf{T}$ and $\mathbf{U}$ has been given through a mapping $\mathcal{M}'$. The schema evolution problem is then to provide a mapping from $\mathbf{S}$ to $\mathbf{U}$, considering the metadata provided by $\mathcal{M}$ and $\mathcal{M}'$. As pointed out in Kolaitis [2005], the process of constructing a schema mapping is time consuming, and thus one would like to solve the schema evolution problem by automatically reusing the metadata that is given. In scenario (a), it is possible to do this by using the composition operator [Fagin et al. 2005; Kolaitis 2005]; the mapping $\mathcal{M} \circ \mathcal{M}'$ correctly represents the relationship between schemas $\mathbf{S}$ and $\mathbf{U}$.

Scenario (b) in Figure 1 is similar to scenario (a), but in this case it has been decided to replace source schema $\mathbf{S}$ by $\mathbf{U}$. As in (a), the relationship between $\mathbf{S}$ and $\mathbf{U}$ is given by a mapping, that is again called $\mathcal{M}'$. The natural question at this point is whether a combination of mappings $\mathcal{M}$ and $\mathcal{M}'$ could be used to provide the right mapping, or at least a good mapping, from $\mathbf{U}$ to $\mathbf{T}$ according to

the metadata provided by $\mathcal{M}$ and $\mathcal{M}'$. It has been argued that the combination of the inverse and composition operators can be used for this purpose, and the mapping $(\mathcal{M}')^{-1} \circ \mathcal{M}$ has been proposed as a solution for the schema evolution problem [Fagin 2007], where $(\mathcal{M}')^{-1}$ represents an inverse of mapping $\mathcal{M}'$. But, unfortunately, it has not been formally studied to what extend $(\mathcal{M}')^{-1} \circ \mathcal{M}$ is the right solution for the schema evolution problem. In this section, we address this issue for the common case of mappings given by st-tgds, and show that if $(\mathcal{M}')^{-1}$ is the maximum recovery of $\mathcal{M}'$, then $(\mathcal{M}')^{-1} \circ \mathcal{M}$ is the best solution in a precise sense for the schema evolution problem.

For the rest of this section, let $\mathbf{S}$ be a source schema, $\mathbf{T}$, and $\mathbf{U}$, target schemas, $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, and $\mathcal{M}' = (\mathbf{S}, \mathbf{U}, \Sigma')$, where Σ and Σ' are sets of st-tgds. If $\mathcal{M}^\star$ is a mapping from $\mathbf{U}$ to $\mathbf{T}$, what properties should it satisfy in order to be considered a good solution for the schema evolution problem? Or, in other words, what properties should $\mathcal{M}^\star$ satisfy to be considered a good representation of the metadata provided by $\mathcal{M}$ and $\mathcal{M}'$? Assume that I is an instance of $\mathbf{S}$, and let J be a solution for I under $\mathcal{M}'$. If J properly represents the information in I, then one would consider $\mathcal{M}^\star$ a good representation of the metadata provided by $\mathcal{M}$ and $\mathcal{M}'$ if the space of solutions for I under $\mathcal{M}$ is the same as the space of solutions for J under $\mathcal{M}^\star$, that is, $\mathrm{Sol}_{\mathcal{M}}(I) = \mathrm{Sol}_{\mathcal{M}^\star}(J)$. Or, at least, one would expect that none of the instances in $\mathrm{Sol}_{\mathcal{M}}(I)$ are ruled out by $\mathcal{M}^\star$ when mapping data from J, that is, $\mathrm{Sol}_{\mathcal{M}}(I) \subseteq \mathrm{Sol}_{\mathcal{M}^\star}(J)$. In this section, we use this simple criterion to compare different solutions for the schema evolution problem.

To formalize this criterion, for every instance I of $\mathbf{S}$, we first need to choose a particular solution J under $\mathcal{M}'$. A natural candidate for this is the canonical universal solution [Fagin et al. 2005a], which has been identified in the database literature as a solution with several desirable properties [Fagin et al. 2005a; Hernich and Schweikardt 2009]. In the following, we show how to compute the canonical universal solution for a source instance I under the schema mapping $\mathcal{M}' = (\mathbf{S}, \mathbf{U}, \Sigma')$. For every st-tgd in Σ' of the form $\varphi(\bar{x}) \rightarrow \exists \bar{y}\, \psi(\bar{x}, \bar{y})$, where $\bar{x} = (x_1, \ldots, x_k)$ and $\bar{y} = (y_1, \ldots, y_\ell)$ are tuples of distinct variables, and for every k-tuple $\bar{a}$ from $\mathrm{dom}(I)$ such that $I \models \varphi(\bar{a})$, do the following. First choose an ℓ-tuple $\bar{n}$ of distinct fresh values from $\mathbf{N}$, and then include all the conjuncts in $\psi(\bar{a}, \bar{n})$ in the canonical universal solution for I. Furthermore, the canonical universal solution only contains tuples that are obtained by applying the previous procedure [Fagin et al. 2005a].

Example 5.1. Assume that $\Sigma' = \{S(x_1, x_2) \rightarrow \exists y_1 \exists y_2 (T(x_1, y_1) \wedge U(x_2, y_2, y_1))\}$ and that I is a source instance such that $S^I = \{(a, b), (c, d)\}$. Given that $I \models S(a, b)$, this procedure chooses a tuple (n_1, n_2) of fresh null values, and then it adds tuples (a, n_1) to T and (b, n_2, n_1) to U in the canonical universal solution J for I under Σ'. In the same way, given that $I \models S(c, d)$, the procedure chooses a tuple (n_3, n_4) of fresh null values, and then it adds tuples (c, n_3) to T^J and (d, n_4, n_3) to U^J. Finally, given that (a, b) and (c, d) are the only tuples from $\mathrm{dom}(I)$ for which I satisfies formula $S(x, y)$, we conclude that $T^J = \{(a, n_1), (c, n_3)\}$ and $U^J = \{(b, n_2, n_1), (d, n_4, n_3)\}$.

It is important to notice that the canonical universal solution for I under $\mathcal{M}'$ corresponds to the naïve chase of I with Σ' (see Section 6 for more details

about the chase procedure). Thus, we use the term $\text{chase}_{\Sigma'}(I)$ to denote the canonical universal solution for I under $\mathcal{M}'$. With this notation, the criterion is formalized as follows: A mapping $\mathcal{M}^\star$ from $\mathbf{U}$ to $\mathbf{T}$ is said to be a *solution for the schema evolution problem for $\mathcal{M}$ and $\mathcal{M}'$* if for every instance I of $\mathbf{S}$, it holds that:

$$\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}^\star}(\text{chase}_{\Sigma'}(I)).$$

The previous criterion also suggests a way to compare alternative solutions for the schema evolution problem; the closer the space of solutions $\text{Sol}_{\mathcal{M}^\star}(\text{chase}_{\Sigma'}(I))$ is to $\text{Sol}_{\mathcal{M}}(I)$ the better is $\mathcal{M}^\star$ as a solution for the schema evolution problem. In the following proposition, we show that under this criterion, the notion of maximum recovery can be used to obtain the best solution for the schema evolution problem.

PROPOSITION 5.2. *Let $\mathbf{S}$ be a source schema, $\mathbf{T}$, $\mathbf{U}$ target schemas, $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, and $\mathcal{M}' = (\mathbf{S}, \mathbf{U}, \Sigma')$, where Σ and Σ' are sets of st-tgds. Then there exists a maximum recovery $\mathcal{N}$ of $\mathcal{M}'$ such that:*

(1) *$\mathcal{N} \circ \mathcal{M}$ is a solution for the schema evolution problem for $\mathcal{M}$ and $\mathcal{M}'$, and*
(2) *for every solution $\mathcal{M}^\star$ for the schema evolution problem for $\mathcal{M}$ and $\mathcal{M}'$, and for every instance I of $\mathbf{S}$, it holds that:*

$$\text{Sol}_{\mathcal{N}\circ\mathcal{M}}(\text{chase}_{\Sigma'}(I)) \subseteq \text{Sol}_{\mathcal{M}^\star}(\text{chase}_{\Sigma'}(I)).$$

PROOF. Let $\mathcal{N} = \{(\text{chase}_{\Sigma'}(I), I) \mid I \in \text{Inst}(\mathbf{S})\}$. Given that $\text{dom}(\mathcal{M}') = \mathbf{S}$, we have that $\mathcal{N}$ is a reduced recovery of $\mathcal{M}'$. Moreover, for every $(J, I) \in \mathcal{N}$, given that $J = \text{chase}_{\Sigma'}(I)$, we have that J is a witness solution for I under $\mathcal{M}'$ by Lemma 4.1 and the fact that $\text{chase}_{\Sigma'}(I)$ is a universal solution for I under $\mathcal{M}'$ [Fagin et al. 2005a; Arenas et al. 2004]. Thus, by Lemma 3.11, we conclude that $\mathcal{N}$ is a maximum recovery of $\mathcal{M}'$. Next we show that $\mathcal{N}$ satisfies the two conditions of the proposition.

(1) For every instance I of $\mathbf{S}$, we have that $(\text{chase}_{\Sigma'}(I), I) \in \mathcal{N}$ and, thus, we conclude that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{N}\circ\mathcal{M}}(\text{chase}_{\Sigma'}(I))$. Thus, we have that $\mathcal{N} \circ \mathcal{M}$ is a solution for the schema evolution problem for $\mathcal{M}$ and $\mathcal{M}'$.

(2) Let $\mathcal{M}^\star$ be a solution for the schema evolution problem for $\mathcal{M}$ and $\mathcal{M}'$, and I an instance of $\mathbf{S}$. We need to show that $\text{Sol}_{\mathcal{N}\circ\mathcal{M}}(\text{chase}_{\Sigma'}(I)) \subseteq \text{Sol}_{\mathcal{M}^\star}(\text{chase}_{\Sigma'}(I))$.

Assume that $J \in \text{Sol}_{\mathcal{N}\circ\mathcal{M}}(\text{chase}_{\Sigma'}(I))$. Then there exists an instance I' of $\mathbf{S}$ such that $(\text{chase}_{\Sigma'}(I), I') \in \mathcal{N}$ and $(I', J) \in \mathcal{M}$. Given that $\mathcal{M}^\star$ is a solution for the schema evolution problem for $\mathcal{M}$ and $\mathcal{M}'$, we have that $\text{Sol}_{\mathcal{M}}(I') \subseteq \text{Sol}_{\mathcal{M}^\star}(\text{chase}_{\Sigma'}(I'))$ and, hence, $J \in \text{Sol}_{\mathcal{M}^\star}(\text{chase}_{\Sigma'}(I'))$. But, by definition of $\mathcal{N}$, we have that $\text{chase}_{\Sigma'}(I') = \text{chase}_{\Sigma'}(I)$ since $(\text{chase}_{\Sigma'}(I), I') \in \mathcal{N}$. Thus, we have that $J \in \text{Sol}_{\mathcal{M}^\star}(\text{chase}_{\Sigma'}(I))$. This concludes the proof of the proposition. □

Notice that an ideal solution for the schema evolution problem for mappings $\mathcal{M}$ and $\mathcal{M}'$ is a mapping $\mathcal{M}^\star$ such that $\text{Sol}_{\mathcal{M}}(I) = \text{Sol}_{\mathcal{M}^\star}(\text{chase}_{\Sigma'}(I))$, for every source instance I. The following corollary of Proposition 5.2 shows that if such a solution exists, then one can focus on the solutions constructed by using maximum recoveries in order to find an ideal solution.

COROLLARY 5.3. *Let* $\mathbf{S}$ *be a source schema,* $\mathbf{T}$, $\mathbf{U}$, *target schemas,* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ *and* $\mathcal{M}' = (\mathbf{S}, \mathbf{U}, \Sigma')$, *with* Σ, Σ' *sets of st-tgds. If there exists an ideal solution for the schema evolution problem for* $\mathcal{M}$ *and* $\mathcal{M}'$, *then there exists a maximum recovery* $\mathcal{N}$ *of* $\mathcal{M}'$ *such that* $\mathcal{N} \circ \mathcal{M}$ *is an ideal solution for the schema evolution problem for* $\mathcal{M}$ *and* $\mathcal{M}'$.

From Proposition 5.2 and the previous corollary, we conclude that the combination of the maximum recovery and the composition operator is appropriate to provide a solution for the schema evolution problem shown in Figure 1 (b). We also note that maximum recovery can be replaced neither by inverse nor by quasi-inverse in Proposition 5.2, as it is known that even for full st-tgds, inverses and quasi-inverses are not guaranteed to exist [Fagin 2007; Fagin et al. 2008].

6. COMPARISON WITH THE NOTIONS OF INVERSE AND QUASI-INVERSE

In this section, we study the relationship between the notion of maximum recovery and the notions of inverse and quasi-inverse [Fagin 2007; Fagin et al. 2008].

We start by recalling the definition of inverse proposed in Fagin [2007]. A mapping $\mathcal{M}$ is *closed-down on the left* if whenever $(I, J) \in \mathcal{M}$ and $I' \subseteq I$, it holds that $(I', J) \in \mathcal{M}$. Fagin [2007] defines a notion of inverse focusing on mappings that satisfy this condition. More precisely, let $\mathbf{S}$ be a source schema. Fagin first defines an identity mapping $\overline{\mathrm{Id}}$ as $\{(I_1, I_2) \mid (I_1, I_2) \in \mathrm{Inst}(\mathbf{S}) \times \mathrm{Inst}(\mathbf{S})$ and $I_1 \subseteq I_2\}$, which is appropriate for closed-down on the left mappings [Fagin 2007]. Then he says that a ts-mapping $\mathcal{M}'$ is an *inverse* of an st-mapping $\mathcal{M}$ if and only if $\mathcal{M} \circ \mathcal{M}' = \overline{\mathrm{Id}}$.

Since it is rare that a schema mapping possesses an inverse, Fagin et al. [2008] introduce the notion of a quasi-inverse of a schema mapping in Fagin et al. [2008]. The idea behind quasi-inverses is to relax the notion of inverse of a mapping by not differentiating between source instances that are data-exchange equivalent. Let $\mathcal{M}$ be a mapping from a source schema $\mathbf{S}$ to a target schema $\mathbf{T}$. Instances I_1 and I_2 of $\mathbf{S}$ are *data-exchange equivalent* with respect to $\mathcal{M}$, denoted by $I_1 \sim_{\mathcal{M}} I_2$, if $\mathrm{Sol}_{\mathcal{M}}(I_1) = \mathrm{Sol}_{\mathcal{M}}(I_2)$. Furthermore, given a mapping $\mathcal{M}_1$ from $\mathbf{S}$ to $\mathbf{S}$, mapping $\mathcal{M}_1[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$ is defined as $\{(I_1, I_2) \in \mathrm{Inst}(\mathbf{S}) \times \mathrm{Inst}(\mathbf{S}) \mid \exists (I_1', I_2') : I_1 \sim_{\mathcal{M}} I_1', I_2 \sim_{\mathcal{M}} I_2'$ and $(I_1', I_2') \in \mathcal{M}_1\}$. Then a ts-mapping $\mathcal{M}'$ is a *quasi-inverse* of an st-mapping $\mathcal{M}$ if $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] = \overline{\mathrm{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$.

The definitions of inverse and quasi-inverse are appropriate for closed-down on the left mappings. In fact, some counterintuitive results are obtained if one removes this restriction. For example, let $\mathbf{S} = \{P(\cdot)\}$, $\mathbf{T} = \{R(\cdot)\}$, and $\mathcal{M}$ be a mapping from $\mathbf{S}$ to $\mathbf{T}$ specified by dependency $\forall x\,(P(x) \leftrightarrow R(x))$. In this case, mapping $\mathcal{M}'$ specified by $\forall x\,(R(x) \leftrightarrow P(x))$ is an ideal inverse of $\mathcal{M}$ since $\mathcal{M} \circ \mathcal{M}' = \mathrm{Id} = \{(I, I) \mid I \in \mathrm{Inst}(\mathbf{S})\}$. However, $\mathcal{M}'$ is neither an inverse nor a quasi-inverse of $\mathcal{M}$ (although it is a maximum recovery of $\mathcal{M}$). Moreover, the definitions of inverse and quasi-inverse are only appropriate for total mappings, that is, mappings $\mathcal{M}$ such that $\mathrm{dom}(\mathcal{M})$ is the set of all source instances. In fact,

as shown in the following proposition, if an st-mapping $\mathcal{M}$ is not total, then $\mathcal{M}$ is neither invertible nor quasi-invertible.

PROPOSITION 6.1. *Let $\mathcal{M}$ be a mapping from a source schema $\mathbf{S}$ to a target schema $\mathbf{T}$. If $\mathcal{M}$ is not a total mapping, then $\mathcal{M}$ is neither invertible nor quasi-invertible.*

PROOF. Assume that $\mathcal{M}$ is not a total mapping, that is, there is an instance I of $\mathbf{S}$ for which $\mathrm{Sol}_{\mathcal{M}}(I) = \emptyset$. We first show that $\mathcal{M}$ is not invertible. By hypothesis, for every mapping $\mathcal{M}' \subseteq \mathrm{Inst}(\mathbf{T}) \times \mathrm{Inst}(\mathbf{S})$, it holds that there is no instance I' such that $(I, I') \in \mathcal{M} \circ \mathcal{M}'$, and, therefore, $\mathcal{M} \circ \mathcal{M}' \neq \overline{\mathrm{Id}}$. Thus, we conclude that $\mathcal{M}$ is not invertible.

Second, we show that $\mathcal{M}$ is not quasi-invertible. Let $\mathcal{M}'$ be a mapping from $\mathbf{T}$ to $\mathbf{S}$. Given that $\mathrm{Sol}_{\mathcal{M}}(I) = \emptyset$, it holds that $\mathrm{Sol}_{(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]}(I) = \emptyset$. In fact, if we assume that $(I, I_1) \in (\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$, then there exist instances I' and I'_1 of $\mathbf{S}$ such that $I \sim_{\mathcal{M}} I'$, $I_1 \sim_{\mathcal{M}} I'_1$ and $(I', I'_1) \in \mathcal{M} \circ \mathcal{M}'$. But if $I \sim_{\mathcal{M}} I'$, then it holds that $\mathrm{Sol}_{\mathcal{M}}(I') = \mathrm{Sol}_{\mathcal{M}}(I) = \emptyset$. Thus, we conclude that $\mathrm{Sol}_{\mathcal{M} \circ \mathcal{M}'}(I') = \emptyset$, which contradicts the fact that $(I', I'_1) \in \mathcal{M} \circ \mathcal{M}'$. On the contrary, we have that $(I, I) \in \overline{\mathrm{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$ since $I \sim_{\mathcal{M}} I$ and $(I, I) \in \overline{\mathrm{Id}}$. Thus, we have that $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] \neq \overline{\mathrm{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$. Therefore, we conclude that $\mathcal{M}$ is not quasi-invertible. □

From the discussion in the previous paragraph, to compare the notions of maximum recovery, inverse, and quasi-inverse, we need to focus on the class of total st-mappings that are closed-down on the left. This class includes, for example, the st-mappings specified by UCQ$^{\neq}$-TO-CQ st-dependencies. Our first result is a corollary of Propositions 3.22 and 3.24 in Fagin et al. [2008] and Theorem 4.2.

PROPOSITION 6.2. *There exists an st-mapping $\mathcal{M}$ specified by a set of full st-tgds that is neither invertible nor quasi-invertible, but has a maximum recovery.*

This result combined with the following theorem, shows that the notion of maximum recovery strictly generalizes the notion of inverse.

THEOREM 6.3. *Let $\mathcal{M}$ be a total st-mapping that is closed-down on the left, and assume that $\mathcal{M}$ is invertible. Then $\mathcal{M}'$ is an inverse of $\mathcal{M}$ if and only if $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$.*

PROOF. ($\Rightarrow$) Let $\mathcal{M}'$ be an inverse of $\mathcal{M}$. Then $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$ if and only if $I_1 \subseteq I_2$ and, thus, $\mathcal{M}'$ is a recovery of $\mathcal{M}$ since $I_1 \subseteq I_1$. Thus, from Proposition 3.8, we know that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$ if and only if for every $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, it is the case that $\emptyset \subsetneq \mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$. But if $I_1 \subseteq I_2$, we immediately conclude that $\emptyset \subsetneq \mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$ since $\mathcal{M}$ is a closed down on the left and total st-mapping.

($\Leftarrow$) Let $\mathcal{M}'$ be a maximum recovery of $\mathcal{M}$. In order to show that $\mathcal{M}'$ is an inverse of $\mathcal{M}$, we need to show that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$ if and only if $I_1 \subseteq I_2$. First, assume that $I_1 \subseteq I_2$. Given that I_2 is a source instance and $\mathcal{M}'$ is a recovery of $\mathcal{M}$, we know that $(I_2, I_2) \in \mathcal{M} \circ \mathcal{M}'$. Thus, given that $\mathcal{M}$ is closed-down on the left, we have that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$. Second, assume that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$.

Given that $\mathcal{M}$ is invertible, there exists an inverse $\mathcal{M}''$ of $\mathcal{M}$. Then $\mathcal{M}''$ is a recovery of $\mathcal{M}$. Thus, given that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$, we have that $\mathcal{M} \circ \mathcal{M}' \subseteq \mathcal{M} \circ \mathcal{M}''$. We infer that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}''$ since $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, which implies that $I_1 \subseteq I_2$ since $\mathcal{M}''$ is an inverse of $\mathcal{M}$. This concludes the proof of the proposition. □

The exact relationship between the notions of quasi-inverse and maximum recovery is shown in the following theorem. It is worth emphasizing that if an st-mapping $\mathcal{M}$ is quasi-invertible, then it admits a maximum recovery and, furthermore, every maximum recovery of $\mathcal{M}$ is also a quasi-inverse of $\mathcal{M}$.

THEOREM 6.4.

(1) *Let $\mathcal{M}$ be a total st-mapping that is closed-down on the left, and assume that $\mathcal{M}$ is quasi-invertible. Then $\mathcal{M}$ has a maximum recovery and, furthermore, $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$ if and only if $\mathcal{M}'$ is a quasi-inverse and a recovery of $\mathcal{M}$.*
(2) *There exists an st-mapping $\mathcal{M}$ specified by a set of st-tgds and a ts-mapping $\mathcal{M}'$ specified by a set of ts-tgds such that, $\mathcal{M}'$ is a quasi-inverse of $\mathcal{M}$ but not a maximum recovery of $\mathcal{M}$.*

To prove the proposition, we need the following lemma. Recall that an st-mapping $\mathcal{M}$ has the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property if, for every pair of source instances I_1, I_2 such that $\mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$, there exist instances I_1' and I_2' such that $I_1 \sim_{\mathcal{M}} I_1'$, $I_2 \sim_{\mathcal{M}} I_2'$ and $I_1' \subseteq I_2'$. An inspection of the proof of Theorem 3.19 in Fagin et al. [2008], reveals that the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property is still a necessary condition for quasi-invertibility for arbitrary st-mappings. We state this result in the following lemma, and we also include a proof here using our notation. In this proof, we use $(I_1, I_2) \sim_{\mathcal{M}} (I_1', I_2')$ to indicate that $I_1 \sim_{\mathcal{M}} I_1'$ and $I_2 \sim_{\mathcal{M}} I_2'$.

LEMMA 6.5. [*Fagin et al.* 2008] *Let $\mathcal{M}$ be an arbitrary st-mapping. If $\mathcal{M}$ is quasi-invertible, then $\mathcal{M}$ has the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property.*

PROOF. Assume that $\mathcal{M}'$ is a quasi-inverse of $\mathcal{M}$, and let (I_1, I_2) be a pair of source instances such that $\mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$. We need to prove that there exist source instances I_1' and I_2' such that $(I_1, I_2) \sim_{\mathcal{M}} (I_1', I_2')$ and $I_1' \subseteq I_2'$. Given that $I_2 \sim_{\mathcal{M}} I_2$ and $I_2 \subseteq I_2$, we have that $(I_2, I_2) \in \overline{\mathrm{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$. Thus, given that $\mathcal{M}'$ is a quasi-inverse of $\mathcal{M}$, we have that $(I_2, I_2) \in (\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$ and, therefore, there exists a pair of instances (I_3, I_4) such that $(I_2, I_2) \sim_{\mathcal{M}} (I_3, I_4)$ and $(I_3, I_4) \in \mathcal{M} \circ \mathcal{M}'$. Then there exists a target instance J such that $(I_3, J) \in \mathcal{M}$ and $(J, I_4) \in \mathcal{M}'$. Now, given that $I_2 \sim_{\mathcal{M}} I_3$ and $(I_3, J) \in \mathcal{M}$, we obtain that $(I_2, J) \in \mathcal{M}$. Since $\mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$, it also holds that $(I_1, J) \in \mathcal{M}$, and then $(I_1, I_4) \in \mathcal{M} \circ \mathcal{M}'$. Thus, given that $(I_1, I_4) \sim_{\mathcal{M}} (I_1, I_4)$, we obtain that $(I_1, I_4) \in (\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$. Therefore, since $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] = \overline{\mathrm{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$, there exists a pair of source instances (I_5, I_6) such that $(I_1, I_4) \sim_{\mathcal{M}} (I_5, I_6)$ and $I_5 \subseteq I_6$. Finally, given that $I_2 \sim_{\mathcal{M}} I_4$ and $I_4 \sim_{\mathcal{M}} I_6$, we have that $(I_1, I_2) \sim_{\mathcal{M}} (I_5, I_6)$. Thus, we conclude that there exists a pair of source instances (I_5, I_6) such that $(I_1, I_2) \sim_{\mathcal{M}} (I_5, I_6)$ and $I_5 \subseteq I_6$. This concludes the proof of the lemma. □

PROOF OF THEOREM 6.4. Recall that $\mathcal{M}'$ is a quasi-inverse of $\mathcal{M}$ if $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] = \overline{\mathrm{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$, that is if the following statements are equivalent for every pair of instances I_1 and I_2:

(a) There are instances I_1' and I_2' such that $(I_1, I_2) \sim_{\mathcal{M}} (I_1', I_2')$ and $I_1' \subseteq I_2'$.
(b) There are instances I_1'' and I_2'' such that $(I_1, I_2) \sim_{\mathcal{M}} (I_1'', I_2'')$ and $(I_1'', I_2'') \in \mathcal{M} \circ \mathcal{M}'$.

Now to prove (1), let $\mathcal{M}$ be an st-mapping that is closed-down on the left, and assume that $\mathcal{M}$ is quasi-invertible. From Theorem 3.12, to prove that $\mathcal{M}$ has a maximum recovery, we need to prove that $\mathcal{M}$ has witness solutions for every source instance. Assume that $\mathcal{M}'$ is a quasi-inverse of $\mathcal{M}$, and let I be an arbitrary source instance. By the definition of quasi-inverse, and given that $(I, I) \sim_{\mathcal{M}} (I, I)$ and $I \subseteq I$ (condition (a)) we know that there exist I_1 and I_2 such that $(I, I) \sim_{\mathcal{M}} (I_1, I_2)$ and $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$ (condition (b)). Then, there is a target instance J such that $(I_1, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$. We claim that J is a witness solution for I. First note that, since $I \sim_{\mathcal{M}} I_1$ and $(I_1, J) \in \mathcal{M}$, we have that $J \in \mathrm{Sol}_{\mathcal{M}}(I)$. Now assume that there is an instance I' such that $J \in \mathrm{Sol}_{\mathcal{M}}(I')$, we must prove that $\mathrm{Sol}_{\mathcal{M}}(I) \subseteq \mathrm{Sol}_{\mathcal{M}}(I')$. Given that $J \in \mathrm{Sol}_{\mathcal{M}}(I')$ and $(J, I_2) \in \mathcal{M}'$, we obtain that $(I', I_2) \in \mathcal{M} \circ \mathcal{M}'$. Thus, $(I', I) \sim_{\mathcal{M}} (I', I_2)$ and $(I', I_2) \in \mathcal{M} \circ \mathcal{M}'$ (condition (b)) and, hence, there exists a pair of source instances (K_1, K_2) such that $(I', I) \sim_{\mathcal{M}} (K_1, K_2)$ and $K_1 \subseteq K_2$ (condition (a)). Given that $\mathcal{M}$ is closed-down on the left, we obtain that $\mathrm{Sol}_{\mathcal{M}}(K_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(K_1)$, and then from $(I', I) \sim_{\mathcal{M}} (K_1, K_2)$ we conclude that $\mathrm{Sol}_{\mathcal{M}}(I) \subseteq \mathrm{Sol}_{\mathcal{M}}(I')$. This was to be shown.

Now, we show that, provided that $\mathcal{M}$ is quasi-invertible, it holds that every maximum recovery of $\mathcal{M}$ is also a quasi-inverse of $\mathcal{M}$. Let $\mathcal{M}'$ be a maximum recovery of $\mathcal{M}$, we have to show that condition (a) holds if and only if condition (b) holds.

(a) $\Rightarrow$ (b): Let I_1 and I_2 be source instances, and assume that there exist instances I_1' and I_2' such that $(I_1, I_2) \sim_{\mathcal{M}} (I_1', I_2')$ and $I_1' \subseteq I_2'$. Given that $\mathcal{M}$ is quasi-invertible, we have that $\mathcal{M}$ is a total st-mapping and, hence, $I_2' \in \mathrm{dom}(\mathcal{M})$. Then given that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$, we have that $(I_2', I_2') \in \mathcal{M} \circ \mathcal{M}'$. Thus, since $\mathcal{M}$ is closed-down on the left and $I_1' \subseteq I_2'$, we obtain $(I_1', I_2') \in \mathcal{M} \circ \mathcal{M}'$, which proves that (b) holds.

(b) $\Rightarrow$ (a): Let I_1 and I_2 be source instances, and assume that $(I_1, I_2) \sim_{\mathcal{M}} (I_1'', I_2'')$ and $(I_1'', I_2'') \in \mathcal{M} \circ \mathcal{M}'$. Given that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$, we have by Proposition 3.8 that $\mathrm{Sol}_{\mathcal{M}}(I_2'') \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1'')$. Thus, given that $(I_1, I_2) \sim_{\mathcal{M}} (I_1'', I_2'')$, we conclude that $\mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$. Now, given that $\mathcal{M}$ is quasi-invertible, by Lemma 6.5 we know that $\mathcal{M}$ satisfies the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property. Then from $\mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$, we obtain that there exist instances I_1' and I_2' such that, $(I_1, I_2) \sim_{\mathcal{M}} (I_1', I_2')$ and $I_1' \subseteq I_2'$, which was to be shown.

It only left to show that, if $\mathcal{M}'$ is both a quasi-inverse and a recovery of $\mathcal{M}$, then $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$. Assume then that $\mathcal{M}'$ is a quasi-inverse and a recovery of $\mathcal{M}$, and let $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$. From Proposition 3.8 and the facts that $\mathcal{M}'$ is a recovery of $\mathcal{M}$ and $\mathcal{M}$ is a total mapping, to prove that $\mathcal{M}'$ is

a maximum recovery of $\mathcal{M}$, we need to show that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. Given that $(I_1, I_2) \sim_{\mathcal{M}} (I_1, I_2)$ and $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$ (condition (b)), there exists a pair (I_1', I_2') such that $(I_1, I_2) \sim_{\mathcal{M}} (I_1', I_2')$ and $I_1' \subseteq I_2'$ (condition (a)). Now, given that $\mathcal{M}$ is closed-down on the left, we obtain that $\text{Sol}_{\mathcal{M}}(I_2') \subseteq \text{Sol}_{\mathcal{M}}(I_1')$ and, therefore, $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ since $(I_1, I_2) \sim_{\mathcal{M}} (I_1', I_2')$.

We now prove statement (2) of Theorem 6.4. Let $\mathcal{M}$ be specified by st-tgds $P(x) \rightarrow T(x)$ and $R(x) \rightarrow T(x)$, and $\mathcal{M}'$ specified by ts-tgd $T(x) \rightarrow P(x)$. $\mathcal{M}'$ is a quasi-inverse for $\mathcal{M}$ (see Fagin et al. [2008]), but $\mathcal{M}'$ is not a maximum recovery of $\mathcal{M}$ given that, for example, for the source instance I such that $R^I = \{a\}$ and $P^I = \emptyset$, we have that $I \in \text{dom}(\mathcal{M})$ but $(I, I) \notin \mathcal{M} \circ \mathcal{M}'$. □

6.1 On Necessary and Sufficient Conditions for the Existence of Inverses and Quasi-Inverses

In Section 3, we identify a necessary and sufficient condition for the existence of maximum recoveries. For the case of the inverse (quasi-inverse), a condition called *subset property* (($\sim_{\mathcal{M}}, \sim_{\mathcal{M}}$)-*subset property*) was identified in Fagin et al. [2008] as necessary and sufficient for testing invertibility (quasi-invertibility), for the case of st-mappings specified by st-tgds. In this section, we first show that the subset property (($\sim_{\mathcal{M}}, \sim_{\mathcal{M}}$)-subset property) is not a sufficient condition for testing invertibility (quasi-invertibility) if one goes beyond st-tgds. Then we show that these conditions can be extended to the class of total and closed-down on the left st-mappings, by combining them with any necessary and sufficient condition for the existence of maximum recoveries.

An st-mapping has the subset property if for every pair of instances I_1, I_2 such that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$, it holds that $I_1 \subseteq I_2$. An st-mapping $\mathcal{M}$ has the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-*subset property* if for every pair of instances I_1, I_2 such that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$, there exist instances I_1' and I_2' such that $I_1 \sim_{\mathcal{M}} I_1'$, $I_2 \sim_{\mathcal{M}} I_2'$ and $I_1' \subseteq I_2'$.

PROPOSITION 6.6. *There exist total and closed-down-on-the-left st-mappings specified by* (1) CQ-TO-CQ$^{\neq}$, *and* (2) CQ-TO-UCQ *dependencies, that satisfy both the subset and* ($\sim_{\mathcal{M}}, \sim_{\mathcal{M}}$)-*subset property and are neither invertible nor quasi-invertible.*

PROOF. We start by showing that there exist st-mappings that are total, closed-down-on-the-left, and specified by CQ-TO-CQ$^{\neq}$, CQ-TO-UCQ dependencies, that satisfy the subset property but are not invertible.

(1) CQ-TO-CQ$^{\neq}$: Let $\mathbf{S} = \{F(\cdot), G(\cdot), H(\cdot)\}$, $\mathbf{T} = \{R(\cdot, \cdot, \cdot, \cdot)\}$ and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, an st-mapping specified by the following set Σ of CQ-TO-CQ$^{\neq}$ st-dependencies:

$$
\begin{aligned}
F(x) &\rightarrow \exists y_1 \exists y_2 \exists y_3 (R(x, y_1, y_2, y_3) \wedge y_1 \neq y_2),\\
G(x) &\rightarrow \exists y_1 \exists y_2 \exists y_3 (R(x, y_1, y_2, y_3) \wedge y_1 \neq y_3),\\
H(x) &\rightarrow \exists y_1 \exists y_2 \exists y_3 (R(x, y_1, y_2, y_3) \wedge y_2 \neq y_3).
\end{aligned}
$$

First, we show that $\mathcal{M}$ satisfies the subset property. Let I_1 and I_2 be source instances. We have to show that, if $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ then $I_1 \subseteq I_2$. Assume then that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. Let n_1, n_2 be two elements in $\mathbf{N}$ such that $n_1 \neq n_2$,

and let J_1 be a target instance such that:

$$R^{J_1} = F^{I_2} \times \{n_1\} \times \{n_2\} \times \{n_1\} \cup (G^{I_2} \cup H^{I_2}) \times \{n_1\} \times \{n_1\} \times \{n_2\}.$$

Clearly $J_1 \in \mathrm{Sol}_{\mathcal{M}}(I_2)$, and then $J_1 \in \mathrm{Sol}_{\mathcal{M}}(I_1)$. Thus, for every $a \in F^{I_1}$, there exists a tuple (a, b_1, b_2, b_3) in R^{J_1} such that $b_1 \neq b_2$, which implies that $a \in F^{I_2}$ (by definition of J_1 and the fact that all the tuples in $(G^{I_2} \cup H^{I_2}) \times \{n_1\} \times \{n_1\} \times \{n_2\}$ do not satisfy condition $b_1 \neq b_2$). We conclude that $F^{I_1} \subseteq F^{I_2}$. Similarly, we can use instances J_2:

$$R^{J_2} = G^{I_2} \times \{n_1\} \times \{n_1\} \times \{n_2\} \cup (F^{I_2} \cup H^{I_2}) \times \{n_1\} \times \{n_2\} \times \{n_1\},$$

and J_3:

$$R^{J_3} = H^{I_2} \times \{n_1\} \times \{n_2\} \times \{n_1\} \cup (F^{I_2} \cup G^{I_2}) \times \{n_1\} \times \{n_2\} \times \{n_2\},$$

to show that $G^{I_1} \subseteq G^{I_2}$ and $H^{I_1} \subseteq H^{I_2}$, respectively. We conclude that $I_1 \subseteq I_2$.

We show now that $\mathcal{M}$ is not invertible. From Theorem 6.3 and the fact that $\mathcal{M}$ is closed-down on the left, to prove that $\mathcal{M}$ is not invertible, it is enough to prove that $\mathcal{M}$ does not have a maximum recovery. Let I_1 be the instance such that $F^{I_1} = \{a\}$ and $G^{I_1} = H^{I_1} = \emptyset$, where a is an arbitrary element of $\mathbf{C}$. Next we show that there is no $J \in \mathrm{Sol}_{\mathcal{M}}(I_1)$ such that J is a witness for I_1, which implies by Theorem 3.12 that $\mathcal{M}$ does not have a maximum recovery. On the contrary, assume that I_1 has a witness solution J_1. Given that $(I_1, J_1) \in \mathcal{M}$, we have that R^{J_1} contains a tuple (a, b_1, b_2, b_3) with $b_1 \neq b_2$. We consider two cases, depending on the values of b_2 and b_3. In the first case we assume that $b_2 = b_3$, and in the second case we assume that $b_1 \neq b_3$. Consider source instance I_2 such that $G^{I_2} = \{a\}$ and $F^{I_2} = H^{I_2} = \emptyset$. We have that $J_1 \in \mathrm{Sol}_{\mathcal{M}}(I_2)$, but it is not the case that $\mathrm{Sol}_{\mathcal{M}}(I_1) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_2)$, which contradicts the fact that J_1 is a witness solution for I_1. Second, assume that $b_2 \neq b_3$, and consider source instance I_3 such that $H^{I_3} = \{a\}$ and $F^{I_3} = G^{I_3} = \emptyset$. We have that $J_1 \in \mathrm{Sol}_{\mathcal{M}}(I_3)$, but it is not the case that $\mathrm{Sol}_{\mathcal{M}}(I_1) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_3)$, which contradicts the fact that J_1 is a witness solution for I_1. This concludes the proof that the mapping $\mathcal{M}$ specified by CQ-TO-CQ$^{\neq}$ st-dependencies is not invertible.

(2) CQ-TO-UCQ: Let $\mathbf{S} = \{F(\cdot), G(\cdot), H(\cdot)\}$, $\mathbf{T} = \{R(\cdot), S(\cdot), T(\cdot)\}$ and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, an st-mapping specified by the following set, Σ, of CQ-TO-UCQ dependencies:

$$\begin{aligned} F(x) &\rightarrow R(x) \vee S(x), \\ G(x) &\rightarrow S(x) \vee T(x), \\ H(x) &\rightarrow T(x) \vee R(x). \end{aligned}$$

We show first that $\mathcal{M}$ satisfies the subset property. Let I_1 and I_2 be source instances. Then we have to show that, if $\mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$, then $I_1 \subseteq I_2$. For the sake of contradiction, assume $\mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$ and $I_1 \not\subseteq I_2$. Then either $F^{I_1} \not\subseteq F^{I_2}$, or $G^{I_1} \not\subseteq G^{I_2}$, or $H^{I_1} \not\subseteq H^{I_2}$. Assume first that $F^{I_1} \not\subseteq F^{I_2}$. Then there exists an element a such that $a \in F^{I_1}$ but $a \notin F^{I_2}$. Let J be a solution for I_2 such that $R^J = F^{I_2}$, $S^J = \emptyset$ and $T^J = G^{I_2} \cup H^{I_2}$. Now, for every solution $J' \in \mathrm{Sol}_{\mathcal{M}}(I_1)$, we have that $a \in R^{J'}$ or $a \in S^{J'}$. Thus, given that $a \notin R^J$ and $S^J = \emptyset$, we obtain that $J \notin \mathrm{Sol}_{\mathcal{M}}(I_1)$, and then $\mathrm{Sol}_{\mathcal{M}}(I_2) \not\subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$,

which contradicts our initial assumption. By using a similar argument, we can show that if $G^{I_1} \not\subseteq G^{I_2}$, then $\mathrm{Sol}_{\mathcal{M}}(I_2) \not\subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$, and if $H^{I_1} \not\subseteq H^{I_2}$, then $\mathrm{Sol}_{\mathcal{M}}(I_2) \not\subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$, which also lead to a contradiction.

In the proof of Proposition 4.8, we show that this set of CQ-TO-UCQ dependencies does not have a maximum recovery. Thus, from Theorem 6.3, we conclude that $\mathcal{M}$ is not invertible.

To conclude the proof of the proposition, we show that examples of CQ-TO-CQ$^{\neq}$ and CQ-TO-UCQ dependencies satisfy the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property but are not quasi-invertible. In this proof, we need the following facts. Let $\mathcal{M}$ be an arbitrary st-mapping. Given that $I \sim_{\mathcal{M}} I$ for every source instance, if $\mathcal{M}$ satisfies the subset property, then $\mathcal{M}$ also satisfies the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property. Furthermore, if $\mathcal{M}$ satisfies the subset property, then $I_1 \sim_{\mathcal{M}} I_2$ if and only if $I_1 = I_2$. Thus, if $\mathcal{M}$ satisfies the subset property, we have that $\mathcal{M}$ has an inverse if and only if $\mathcal{M}$ has a quasi-inverse.

We now prove that both examples satisfy the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property but are not quasi-invertible. We know that both mappings satisfy the subset property, which by the previous discussion implies that both mappings satisfy the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property. Furthermore, we also know that both mappings are not invertible, and therefore, they are not quasi-invertible by the previous discussion and the fact that both mappings satisfy the subset property. This concludes the proof of the proposition. □

It turns out that by using the machinery developed for maximum recoveries, it is possible to provide necessary and sufficient conditions for the existence of inverses and quasi-inverses.

PROPOSITION 6.7. *Let $\mathcal{M}$ be a total st-mapping that is closed-down on the left.*

(1) *$\mathcal{M}$ is invertible if and only if $\mathcal{M}$ has a maximum recovery and satisfies the subset property.*
(2) *$\mathcal{M}$ is quasi-invertible if and only if $\mathcal{M}$ has a maximum recovery and satisfies the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property.*

The proposition is a corollary of Lemmas 6.8 and 6.9.

LEMMA 6.8. *Let $\mathcal{M}$ be a total st-mapping that is closed-down on the left. The following statements are equivalent:*

(1) *$\mathcal{M}$ is quasi-invertible.*
(2) *$\mathcal{M}$ has a maximum recovery and satisfies the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property.*
(3) *For every source instance I_1, there exists $J \in \mathrm{Sol}_{\mathcal{M}}(I_1)$ such that, for every instance I_2 such that $J \in \mathrm{Sol}_{\mathcal{M}}(I_2)$, there exists a pair $(I_1', I_2') \sim_{\mathcal{M}} (I_1, I_2)$ such that $I_2' \subseteq I_1'$.*

PROOF. (1) $\Rightarrow$ (2). It follows directly from Lemma 6.5 and Theorem 6.4.

(2) $\Rightarrow$ (3). Assume that $\mathcal{M}$ has a maximum recovery. Then by Theorem 3.12, we have that $\mathcal{M}$ has witness solutions for every source instance (note that $\mathcal{M}$ is a total mapping). That is, for every source instance I_1, there exists a target instance $J \in \mathrm{Sol}_{\mathcal{M}}(I_1)$, such that for every source instance I_2, such that

$J \in \mathrm{Sol}_{\mathcal{M}}(I_2)$, it is the case that $\mathrm{Sol}_{\mathcal{M}}(I_1) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_2)$. Thus, given that $\mathcal{M}$ satisfies the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property, we obtain that there exists (I_1', I_2') such that $(I_1, I_2) \sim_{\mathcal{M}} (I_1', I_2')$ and $I_2' \subseteq I_1'$. We conclude that (3) holds.

(3) $\Rightarrow$ (1). For every source instance I, let U_I be the set of all target instances $J \in \mathrm{Sol}_{\mathcal{M}}(I)$ that satisfy condition (3), and $\mathcal{M}' = \{(J, I) \mid I$ is a source instance and $J \in U_I\}$. Notice that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$ for every source instance I. Next we show that $\mathcal{M}'$ is a quasi-inverse of $\mathcal{M}$, that is we show that $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] = \overline{\mathrm{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$. Let (I_1, I_2) be an arbitrary pair of source instances. First, assume that there exists a pair (I_1', I_2') such that $(I_1, I_2) \sim_{\mathcal{M}} (I_1', I_2')$ and $I_1' \subseteq I_2'$. We need to show that there exists a pair (I_1'', I_2'') such that $(I_1, I_2) \sim_{\mathcal{M}} (I_1'', I_2'')$ and $(I_1'', I_2'') \in \mathcal{M} \circ \mathcal{M}'$. Given that $\mathcal{M}$ is closed-down on the left, $(I_2', I_2') \in \mathcal{M} \circ \mathcal{M}'$ and $I_1' \subseteq I_2'$, we conclude that $(I_1', I_2') \in \mathcal{M} \circ \mathcal{M}'$ and, therefore we can take I_1'' to be I_1' and I_2'' to be I_2'. Second, assume that there exists a pair (K_1, K_2) such that $(I_1, I_2) \sim_{\mathcal{M}} (K_1, K_2)$ and $(K_1, K_2) \in \mathcal{M} \circ \mathcal{M}'$. We need to prove that there exists a pair (K_1', K_2') such that $(I_1, I_2) \sim_{\mathcal{M}} (K_1', K_2')$ and $K_1' \subseteq K_2'$. Given that $(K_1, K_2) \in \mathcal{M} \circ \mathcal{M}'$, there exists a J such that $(K_1, J) \in \mathcal{M}$ and $(J, K_2) \in \mathcal{M}'$. Thus, by definition of $\mathcal{M}'$, we have that $J \in U_{K_2}$. Hence, given that $J \in \mathrm{Sol}_{\mathcal{M}}(K_1)$, we obtain that there exists a pair (K_1', K_2') such that $(K_1, K_2) \sim_{\mathcal{M}} (K_1', K_2')$ and $K_1' \subseteq K_2'$. Therefore, from the fact that $(I_1, I_2) \sim_{\mathcal{M}} (K_1, K_2)$, we conclude that $(I_1, I_2) \sim_{\mathcal{M}} (K_1', K_2')$ and $K_1' \subseteq K_2'$. This concludes the proof of the lemma. □

LEMMA 6.9. *Let $\mathcal{M}$ be a total st-mapping that is closed-down on the left. The following statements are equivalent:*

(1) *$\mathcal{M}$ is invertible.*
(2) *$\mathcal{M}$ has a maximum recovery and satisfies the subset property.*
(3) *For every source instance I_1, there exists $J \in \mathrm{Sol}_{\mathcal{M}}(I_1)$, such that for every instance I_2 such that $J \in \mathrm{Sol}_{\mathcal{M}}(I_2)$, it holds that $I_2 \subseteq I_1$.*

PROOF. (1) $\Rightarrow$ (2). Assume that $\mathcal{M}$ is invertible, and let $\mathcal{M}'$ be an inverse of $\mathcal{M}$. By Theorem 6.3, we know that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$. It remains to prove that $\mathcal{M}$ satisfies the subset property. Suppose that $\mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$, then we need to prove that $I_1 \subseteq I_2$. Given that $(I_2, I_2) \in \mathcal{M} \circ \mathcal{M}'$, there exists a target instance J such that $(I_2, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$. Thus, given that $\mathrm{Sol}_{\mathcal{M}}(I_2) \subseteq \mathrm{Sol}_{\mathcal{M}}(I_1)$, we have that (I_1, J) in $\mathcal{M}$. We conclude that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, which implies that $I_1 \subseteq I_2$ since $\mathcal{M}'$ is an inverse of $\mathcal{M}$.

(2) $\Rightarrow$ (3). As we pointed out in the proof of Proposition 6.6, if $\mathcal{M}$ satisfies the subset property, then $\mathcal{M}$ satisfies the $(\sim_{\mathcal{M}}, \sim_{\mathcal{M}})$-subset property and for every pair of source instances I_1, I_2, it holds that $I_1 \sim_{\mathcal{M}} I_2$ if and only if $I_1 = I_2$. Thus, (2) $\Rightarrow$ (3) is a direct consequence of the implication (2) $\Rightarrow$ (3) of Lemma 6.8.

(3) $\Rightarrow$ (1). For every source instance I, let U_I be the set of all target instances $J \in \mathrm{Sol}_{\mathcal{M}}(I)$ that satisfy condition (3), and let $\mathcal{M}' = \{(J, I) \mid I$ is a source instance and $J \in U_I\}$. Notice that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$ for every source instance I. Next we show that $\mathcal{M}'$ is an inverse of $\mathcal{M}$, that is we show that for every pair of source instances I_1, I_2, it holds that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$ if and only if $I_1 \subseteq I_2$. First, assume that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$. Then there exists a target instance J such that $(I_1, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$. By definition of $\mathcal{M}'$, we have that $J \in U_{I_2}$. Thus, given that $J \in \mathrm{Sol}_{\mathcal{M}}(I_1)$, we obtain that $I_1 \subseteq I_2$. Second, assume that

$I_1 \subseteq I_2$. Then given that $(I_2, I_2) \in \mathcal{M} \circ \mathcal{M}'$ and $\mathcal{M}$ is closed-down on the left, we obtain that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$. This concludes the proof of the lemma. □

As a corollary of Lemma 6.9, we obtain that an extension of the notion of witness solution can be used to provide a necessary and sufficient condition for invertibility. Given an st-mapping $\mathcal{M}$, we say that a target instance J is a *strong witness* for a source instance I under $\mathcal{M}$, if for every source instance I' such that $J \in \text{Sol}_{\mathcal{M}}(I')$, it holds that $I' \subseteq I$. Notice that if a mapping $\mathcal{M}$ is closed-down on the left and J is a strong witness for I, then J is a witness for I.

COROLLARY 6.10. *A total and closed-down-on-the-left st-mapping $\mathcal{M}$ is invertible if and only if every source instance has a strong witness solution under $\mathcal{M}$.*

7. COMPUTING MAXIMUM RECOVERIES

In Section 4.1, we show that every st-mapping specified by a set of FO-TO-CQ dependencies has a maximum recovery, but up to this point we have not said anything about the language needed to express it. In this section, we show that every st-mapping specified by a set of FO-TO-CQ dependencies has a maximum recovery specified by a set of CQ^{C}-TO-FO dependencies. In fact, we provide an algorithm that computes maximum recoveries for st-mappings specified by FO-TO-CQ dependencies. Our algorithm runs in exponential time when mappings are given by sets of FO-TO-CQ dependencies, and can be adapted to run in quadratic time when the input is a mapping specified by a set of full FO-TO-CQ dependencies.

7.1 Preliminaries

In this section, we introduce the basic terminology used in our algorithm, and we also present some results that are important in its formulation.

Our algorithm is based on *query rewriting*, and thus, we start by reviewing some basic results about it. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be an st-mapping such that Σ is a set of FO-TO-CQ dependencies, and let Q be a query over schema $\mathbf{T}$. Given a source instance I, the set of *certain answers of Q over I under $\mathcal{M}$*, is the set of tuples that belong to the evaluation of Q over every possible solution for I under $\mathcal{M}$. We denote this set by $\text{certain}_{\mathcal{M}}(Q, I)$. Thus,

$$\text{certain}_{\mathcal{M}}(Q, I) = \bigcap_{J \in \text{Sol}_{\mathcal{M}}(I)} Q(J).$$

Then a query Q' is said to be a *rewriting of Q over the source* if Q' is a query over $\mathbf{S}$ such that for every $I \in \text{Inst}(\mathbf{S})$, it holds that $Q'(I) = \text{certain}_{\mathcal{M}}(Q, I)$. That is, to obtain the set of certain answers of Q over I under $\mathcal{M}$, we just have to evaluate its rewriting Q' over instance I.

The computation of a rewriting of a conjunctive query is a basic step in the algorithm presented in this section. This problem has been extensively studied in the database area [Levy et al. 1995; Abiteboul and Duschka 1998] and, in particular, in the data integration context [Halevy 2000, 2001; Lenzerini 2002]. In particular, the class of CQ-TO-CQ dependencies corresponds to the

class of GLAV mappings in the data integration context [Lenzerini 2002], and as such, the techniques developed to solve the query-rewriting problem for GLAV mappings can be reused in our context. It is important to notice that most of the query rewriting techniques have been developed for two subclasses of GLAV mappings, namely GAV mappings, which essentially correspond to the class of mappings specified by full CQ-TO-CQ dependencies [Lenzerini 2002], and LAV mappings, which are mappings specified by CQ-TO-CQ dependencies of the form $R(x_1, \ldots, x_k) \rightarrow \psi(x_1, \ldots, x_k)$, where R is a source predicate [Lenzerini 2002]. However, it is possible to reuse a large part of the work in this area, since a GLAV mapping can be represented as the composition of a GAV and a LAV mapping.

Example 7.1. Assume that $\mathcal{M}$ is specified by dependency:

$$R(x) \wedge S(x) \rightarrow \exists y\, T(x, y).$$

Then $\mathcal{M}$ is equivalent to the composition of a GAV mapping specified by dependency $R(x) \wedge S(x) \rightarrow U(x)$ and a LAV mapping specified by dependency $U(x) \rightarrow \exists y\, T(x, y)$, where U is an auxiliary relation.

More formally, let $\mathcal{M}$ be a mapping specified by a set of CQ-TO-CQ dependencies and Q a conjunctive query over the target of $\mathcal{M}$. Then one can obtain a rewriting of Q over the source as follows. First, one constructs, as in the previous example, a GAV mapping $\mathcal{M}_1$ and a LAV mapping $\mathcal{M}_2$, such that $\mathcal{M} = \mathcal{M}_1 \circ \mathcal{M}_2$. Second, one obtains a rewriting Q' of Q over the source of $\mathcal{M}_2$ by adopting one of the algorithms proposed in the literature for query rewriting for LAV mappings [Levy et al. 1996; Duschka and Genesereth 1997; Pottinger and Halevy 2001]. Finally, one obtains a rewriting Q'' of Q' over the source of $\mathcal{M}_1$, which is the desired rewriting of Q, by simply unfolding Q' according to the dependencies of mapping $\mathcal{M}_1$ [Lenzerini 2002].

It should be noticed that the time complexity of the rewriting procedure is exponential in the size of the mapping and the query, and that this procedure can also be used for the case of mappings specified by FO-TO-CQ dependencies. If $\mathcal{M}$ is specified by a set of FO-TO-CQ dependencies, then by using the same idea as in Example 7.1, it is possible to show that $\mathcal{M}$ is equivalent to the composition of a mapping $\mathcal{M}_1$ specified by a set of full FO-TO-CQ dependencies and a LAV mapping $\mathcal{M}_2$. Thus, given that the query unfolding process can be carried out over a set of full FO-TO-CQ dependencies in the same way as for GAV mappings, the process described here can be used to compute in exponential time, the rewriting of a target conjunctive query over the source of $\mathcal{M}$.

For the sake of completeness, in this article we propose a novel exponential-time algorithm, that given a mapping $\mathcal{M}$ specified by a set of FO-TO-CQ st-dependencies and a conjunctive query Q over the target schema, produces a rewriting of Q over the source of $\mathcal{M}$. This algorithm does not follow the approach described here as it directly uses the dependencies specifying $\mathcal{M}$ to construct a query rewriting (it does not decompose $\mathcal{M}$ into the composition of two mappings). In particular, the time complexity of the algorithm is exponential, so it could be used as an alternative query rewriting algorithm.

LEMMA 7.2. *There exists an algorithm* QUERYREWRITING *that, given an st-mapping* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$*, with* Σ *a set of* FO-TO-CQ *dependencies, and a conjunctive query* Q *over schema* $\mathbf{T}$*, computes a domain-independent* FO *query* Q' *that is a rewriting of* Q *over the source. The algorithm runs in exponential time and its output is of exponential size in the size of* Σ *and* Q*.*

PROOF. The proof of the lemma is given in electronic Appendix A.1. □

Another notion that would be used in the proof of correctness of our algorithm (and also in other proofs in the following sections), is the notion of *chase*. This notion is tightly related with certain answers and rewriting of queries [Fagin et al. 2005a; Arenas et al. 2004]. Assume that $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is an st-mapping, where Σ is a set of FO-TO-CQ dependencies. Let I be an instance of $\mathbf{S}$, and let J_I be an instance of $\mathbf{T}$ constructed as follows. For every dependency $\sigma \in \Sigma$ of the form $\varphi(\bar{x}) \rightarrow \exists \bar{y}\, \psi(\bar{x}, \bar{y})$, with $\bar{x} = (x_1, \ldots, x_m)$, $\bar{y} = (y_1, \ldots, y_\ell)$ tuples of distinct variables, and for every m-tuple $\bar{a}$ of elements from $\mathrm{dom}(I)$ such that $I \models \varphi(\bar{a})$, do the following. Choose an ℓ-tuple $\bar{n}$ of distinct fresh values from $\mathbf{N}$, and include all the conjuncts of $\psi(\bar{a}, \bar{n})$ in J_I. We call instance J_I *the chase of* I *with* Σ, and write $J_I = \mathrm{chase}_\Sigma(I)$. In Fagin et al. [2005a], the authors prove several properties of $\mathrm{chase}_\Sigma(I)$. In particular, the authors show that if Q is a conjunctive query over $\mathbf{T}$, then the set of certain answers of I under $\mathcal{M}$ is equal to the set of tuples in $Q(\mathrm{chase}_\Sigma(I))$ that only contains constant values. We denote this last set of tuples by $Q(\mathrm{chase}_\Sigma(I))_\downarrow$. Thus, we have that $\mathrm{certain}_\mathcal{M}(Q, I) = Q(\mathrm{chase}_\Sigma(I))_\downarrow$. Notice that if Q' is a rewriting over the source of a conjunctive query Q, then it holds that $Q'(I) = Q(\mathrm{chase}_\Sigma(I))_\downarrow$.

7.2 Computing Maximum Recoveries in the General Case

In this section, we propose an algorithm that, given a mappings $\mathcal{M}$ specified by a set of FO-TO-CQ dependencies, returns a maximum recovery of $\mathcal{M}$.

It is known that the simple process of reversing the arrows of source-to-target dependencies does not necessarily produce inverses, since conclusions of different dependencies may be related [Fagin 2007], since conclusion of a dependency may be implied by the conclusions of other dependencies. The algorithm presented in this section first searches for these relations among conclusions of dependencies, and then suitably composes the premises of related dependencies and reverses the arrows to obtain a maximum recovery. Let us give some intuition with an example. Consider a mapping $\mathcal{M}$ specified by the FO-TO-CQ dependencies:

$$\varphi_1(x_1, x_2) \rightarrow \exists v(P(x_1, v) \wedge R(v, x_2)), \tag{4}$$

$$\varphi_2(y_1, y_2) \rightarrow P(y_1, y_2) \tag{5}$$

$$\varphi_3(z_1, z_2) \rightarrow {}_1, z_2), \tag{6}$$

where φ_1, φ_2, and φ_3 are arbitrary FO formulas with two free variables. In this case, the conjunction of the conclusions of (5) and (6) implies the conclusion of (4) when y_2 is equal to z_1 and both are existentially quantified. The idea behind the algorithm is to make explicit these types of relationships. For instance, we

could replace (4) by the dependency:

$$\varphi_1(u_1, u_2) \ \vee \ \exists y_2 \exists z_1 \big(\varphi_2(u_1, y_2) \wedge \varphi_3(z_1, u_2) \wedge y_2 = z_1\big) \ \rightarrow \ \exists v(P(u_1, v) \wedge R(v, u_2)). \quad (7)$$

It can be proved that the set of dependencies obtained by replacing formula (4) by (7) is logically equivalent to the initial set of dependencies. After making explicit these types of relationships between dependencies, the algorithm reverses the arrows to obtain a maximum recovery. When reversing the arrows, we also need to impose an additional constraint. In this example, given that (4) is a non-full dependency, when reversing (5), the algorithm needs to force variable y_2 in $P(y_1, y_2)$ to take values only from the set $\mathbf{C}$, that is, we have to use dependency $P(y_1, y_2) \wedge \mathbf{C}(y_2) \rightarrow \varphi_2(y_1, y_2)$ instead of $P(y_1, y_2) \rightarrow \varphi_2(y_1, y_2)$. This is because, given a source instance I such that $I \models \varphi_1(a, b)$, dependency (4) could be satisfied by including a tuple of the form $P(a, n)$ in a target instance, where $n \in \mathbf{N}$, and value n should not be passed to a source instance by a recovery (see Proposition 8.1 for a formal justification for the use of predicate $\mathbf{C}(\cdot)$). In fact, as a safety condition, the algorithm presented in this section uses predicate $\mathbf{C}(\cdot)$ over each variable that passes values from the target to the source. Summing up, the following set of dependencies defines a maximum recovery of the mapping $\mathcal{M}$ above:

$$\begin{aligned} P(y_1, y_2) \wedge \mathbf{C}(y_1) \wedge \mathbf{C}(y_2) &\rightarrow \varphi_2(y_1, y_2) \\ R(z_1, z_2) \wedge \mathbf{C}(z_1) \wedge \mathbf{C}(z_2) &\rightarrow \varphi_3(z_1, z_2), \\ \exists v(P(u_1, v) \wedge R(v, u_2)) \wedge \mathbf{C}(u_1) \wedge \mathbf{C}(u_2) &\rightarrow \varphi_1(u_1, u_2) \ \vee \ \exists y_2 \exists z_1 \\ & \quad (\varphi_2(u_1, y_2) \wedge \varphi_3(z_1, u_2) \wedge y_2 = z_1). \end{aligned}$$

The following algorithm uses a query rewriting procedure to find the types of relationships between these dependencies. In fact, in the example, the formula:

$$\varphi_1(u_1, u_2) \ \vee \ \exists y_2 \exists z_1 \big(\varphi_2(u_1, y_2) \wedge \varphi_3(z_1, u_2) \wedge y_2 = z_1\big), \quad (8)$$

that appears as the premise of (7), makes explicit the relationship between the conclusion $\exists v(P(u_1, v) \wedge R(v, u_2))$ of FO-TO-CQ dependency (4) and dependencies (4), (5), and (6). But not only that, it can be shown that (8) is a rewriting of $\exists v(P(u_1, v) \wedge R(v, u_2))$ over the source schema (according to dependencies (4), (5) and (6)).

In the algorithm, if $\bar{x} = (x_1, \ldots, x_k)$, then $\mathbf{C}(\bar{x})$ is a shorthand for $\mathbf{C}(x_1) \wedge \cdots \wedge \mathbf{C}(x_k)$.

Algorithm MAXIMUMRECOVERY($\mathcal{M}$)

Input: An st-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ is a set of FO-TO-CQ dependencies.
Output: A ts-mapping $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$, where Σ' is a set of $\text{CQ}^{\mathbf{C}}$-TO-FO dependencies and $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$.

(1) Start with Σ' as the empty set.
(2) For every dependency $\sigma \in \Sigma$ of the form $\varphi(\bar{x}) \rightarrow \exists \bar{y}\, \psi(\bar{x}, \bar{y})$, do the following:
  (a) Let Q be the conjunctive query defined by $\exists \bar{y}\, \psi(\bar{x}, \bar{y})$.
  (b) Use QUERYREWRITING($\mathcal{M}$, Q) to compute an FO formula $\alpha(\bar{x})$ that is a rewriting of $\exists \bar{y}\, \psi(\bar{x}, \bar{y})$ over the source.

(c) Add dependency $\exists \bar{y}\, \psi(\bar{x}, \bar{y}) \wedge \mathbf{C}(\bar{x}) \rightarrow \alpha(\bar{x})$ to Σ'.
(3) Return $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$.

THEOREM 7.3. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be an st-mapping, where Σ is a set of* FO-TO-CQ *dependencies. Then* MAXIMUMRECOVERY($\mathcal{M}$) *computes a maximum recovery of $\mathcal{M}$ in exponential time in the size of Σ, which is specified by a set of* $\text{CQ}^{\mathbf{C}}$-TO-FO *dependencies.*

PROOF. From Lemma 7.2, it is straightforward to conclude that the algorithm runs in exponential time. Assume that $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$ is the output of MAXIMUMRECOVERY($\mathcal{M}$). We first show that $\mathcal{M}'$ is a recovery of $\mathcal{M}$, that is, we show that for every instance I of $\mathbf{S}$, it holds that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$.

We show now that $(\text{chase}_\Sigma(I), I) \in \mathcal{M}'$ and, thus, since $(I, \text{chase}_\Sigma(I)) \in \mathcal{M}$, we obtain that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$. Let $\sigma' \in \Sigma'$, we need to show that $(\text{chase}_\Sigma(I), I) \models \sigma'$. Assume that σ' is of the form $\exists \bar{y}\, \psi(\bar{x}, \bar{y}) \wedge \mathbf{C}(\bar{x}) \rightarrow \alpha(\bar{x})$, and that $\bar{a}$ is a tuple of values such that $\text{chase}_\Sigma(I) \models \exists \bar{y}\, \psi(\bar{a}, \bar{y}) \wedge \mathbf{C}(\bar{a})$. We have to show that $I \models \alpha(\bar{a})$. Now, consider the conjunctive query Q_ψ defined by formula $\exists \bar{y}\, \psi(\bar{x}, \bar{y})$. Since $\mathbf{C}(\bar{a})$ holds and $\text{chase}_\Sigma(I) \models \exists \bar{y}\, \psi(\bar{a}, \bar{y})$, we obtain that $\bar{a} \in Q_\psi(\text{chase}_\Sigma(I))_\downarrow$. Thus, by the properties of the chase, we know that $\bar{a} \in \text{certain}_\mathcal{M}(Q_\psi, I)$. Consider now the query Q_α defined by formula $\alpha(\bar{x})$. By the definition of MAXIMUMRECOVERY, we know that Q_α is a rewriting of Q_ψ over schema $\mathbf{S}$, and then $\text{certain}_\mathcal{M}(Q_\psi, I) = Q_\alpha(I)$. Thus, we have that $\bar{a} \in Q_\alpha(I)$, and then $I \models \alpha(\bar{a})$ which was to be shown.

To complete the proof, we show that if $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, then $\emptyset \subsetneq \text{Sol}_\mathcal{M}(I_2) \subseteq \text{Sol}_\mathcal{M}(I_1)$. Thus, by Proposition 3.8 and since $\mathcal{M}'$ is a recovery of $\mathcal{M}$, we obtain that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$. Let $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, and $J^\star$ an instance of $\mathbf{T}$ such that $(I_1, J^\star) \in \mathcal{M}$ and $(J^\star, I_2) \in \mathcal{M}'$. Notice first that $\text{dom}(\mathcal{M}) = \text{Inst}(\mathbf{S})$, and then $\emptyset \subsetneq \text{Sol}_\mathcal{M}(I_2)$. Therefore, we only have to prove that $\text{Sol}_\mathcal{M}(I_2) \subseteq \text{Sol}_\mathcal{M}(I_1)$. Let $J \in \text{Sol}_\mathcal{M}(I_2)$, we need to show that $J \in \text{Sol}_\mathcal{M}(I_1)$. Let $\sigma \in \Sigma$ be a dependency of the form $\varphi(\bar{x}) \rightarrow \exists \bar{y}\, \psi(\bar{x}, \bar{y})$, and assume that $I_1 \models \varphi(\bar{a})$ for some tuple $\bar{a}$ of constant values. We show next that $J \models \exists \bar{y}\, \psi(\bar{a}, \bar{y})$. Since $I_1 \models \varphi(\bar{a})$ we know that for every $J' \in \text{Sol}_\mathcal{M}(I_1)$, it holds that $J' \models \exists \bar{y}\, \psi(\bar{a}, \bar{y})$. In particular, it holds that $J^\star \models \exists \bar{y}\, \psi(\bar{a}, \bar{y})$. By the definition of the algorithm, we know that there exists a dependency $\exists \bar{y}\, \psi(\bar{x}, \bar{y}) \wedge \mathbf{C}(\bar{x}) \rightarrow \alpha(\bar{x})$ in Σ', such that $\alpha(\bar{x})$ is a rewriting of $\exists \bar{y}\, \psi(\bar{x}, \bar{y})$ over $\mathbf{S}$. Then since $J^\star \models \exists \bar{y}\, \psi(\bar{a}, \bar{y})$, $\bar{a}$ is a tuple of constant values, and $(J^\star, I_2) \models \Sigma'$, we know that $I_2 \models \alpha(\bar{a})$. Now consider the queries Q_ψ and Q_α defined by formulas $\exists \bar{y}\, \psi(\bar{x}, \bar{y})$ and $\alpha(\bar{x})$, respectively. Since $I_2 \models \alpha(\bar{a})$, we know that $\bar{a} \in Q_\alpha(I_2)$. Furthermore, we know that $Q_\alpha(I_2) = \text{certain}_\mathcal{M}(Q_\psi, I_2)$, and then $\bar{a} \in \text{certain}_\mathcal{M}(Q_\psi, I_2)$. In particular, since $J \in \text{Sol}_\mathcal{M}(I_2)$, we know that $\bar{a} \in Q_\psi(J)$, from which we conclude that $J \models \exists \bar{y}\, \psi(\bar{a}, \bar{y})$. We have shown that for every $\sigma \in \Sigma$ of the form $\varphi(\bar{x}) \rightarrow \exists \bar{y}\, \psi(\bar{x}, \bar{y})$, if $I_1 \models \varphi(\bar{a})$ for some tuple $\bar{a}$, then $J \models \exists \bar{y}\, \psi(\bar{a}, \bar{y})$. Thus, we have that $(I_1, J) \models \Sigma$ and therefore $J \in \text{Sol}_\mathcal{M}(I_1)$. This concludes the proof of the theorem. □

From Theorems 6.3 and 6.4, we have that if Σ is an invertible (quasi-invertible) set of st-tgds, then MAXIMUMRECOVERY computes an inverse

(quasi-inverse) of Σ. In Fagin et al. [2008], algorithms for computing inverses and quasi-inverses are proposed for the case of mappings given by st-tgds. It is important to note that our algorithm works not only for st-tgds but also for the larger class of FO-TO-CQ dependencies. For the latter class, it is not clear how to extend the algorithms from Fagin et al. [2008] to produce inverses and quasi-inverses, as the notion of generator used in these algorithms (Definition 4.2 in Fagin et al. [2008]) becomes undecidable for FO-TO-CQ dependencies.

The next lemma shows that when the input of algorithm MAXIMUMRECOVERY is a mapping $\mathcal{M}$ specified by a set of st-tgds, then its output is a maximum recovery of $\mathcal{M}$ specified by a set of $\text{CQ}^{\mathbf{C}}$-TO-$\text{UCQ}^{=}$ dependencies. The proof of the lemma follows directly from the proof of Lemma 7.2.

LEMMA 7.4. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be an st-mapping such that Σ is a set of st-tgds, and Q a conjunctive query over schema $\mathbf{T}$. Then algorithm* QUERY REWRITING$(\mathcal{M}, Q)$ *in Lemma* 7.2 *has as output a query Q' in* $\text{UCQ}^{=}$ *that is a rewriting of Q over the source.*

Thus, if the input of our algorithm is a mapping given by a set Σ of st-tgds, it computes a maximum recovery given by a set Σ' of $\text{CQ}^{\mathbf{C}}$-TO-$\text{UCQ}^{=}$ dependencies. In general, the set Σ' computed by our algorithm could be of exponential size in the size of Σ. The following result shows that this exponential blow-up could not be avoided.

THEOREM 7.5. *There exists a family of st-mappings $\{\mathcal{M}_n = (\mathbf{S}_n, \mathbf{T}_n, \Sigma_n)\}_{n\geq 1}$, such that Σ_n is a set of st-tgds of size linear in n, and every set Σ' of* $\text{CQ}^{\mathbf{C}}$-TO-$\text{UCQ}^{=}$ *ts-dependencies that specifies a maximum recovery of $\mathcal{M}_n$ is of size $\Omega(2^n)$.*

PROOF. Let $\mathbf{S}_n = \{R(\cdot), A_1(\cdot), B_1(\cdot), \ldots, A_n(\cdot), B_n(\cdot)\}$, $\mathbf{T}_n = \{P_1(\cdot), \ldots, P_n(\cdot)\}$, and Σ_n the set of st-tgds:

$$\begin{aligned} R(x) &\rightarrow \exists y(P_1(y) \wedge \cdots \wedge P_n(y)), \\ A_1(x) &\rightarrow P_1(x), \\ B_1(x) &\rightarrow P_1(x), \\ &\vdots \\ A_n(x) &\rightarrow P_n(x), \\ B_n(x) &\rightarrow P_n(x). \end{aligned}$$

Let $\mathcal{M}_n = (\mathbf{S}_n, \mathbf{T}_n, \Sigma_n)$ and assume that $\mathcal{M}' = (\mathbf{T}_n, \mathbf{S}_n, \Sigma')$ is a maximum recovery of $\mathcal{M}_n$, where Σ' is a set of $\text{CQ}^{\mathbf{C}}$-TO-$\text{UCQ}^{=}$ ts-dependencies. We first prove some facts about Σ'. Through the proof, we let a be a fixed element in $\mathbf{C}$, and I_R a source instance, such that $R^{I_R} = \{a\}$ and $A_i^{I_R} = B_i^{I_R} = \emptyset$ for every $i \in \{1, \ldots, n\}$. Since $\mathcal{M}'$ is a recovery of $\mathcal{M}_n$, we have that $(I_R, I_R) \in \mathcal{M}_n \circ \mathcal{M}'$. Thus, there exists an instance $J^\star$ such that $(I_R, J^\star) \models \Sigma_n$ and $(J^\star, I_R) \models \Sigma'$. We show first that the domain of $J^\star$ is composed only by null values. On the contrary, assume that there exists a constant element $b \in \mathbf{C}$ such that $b \in \text{dom}(J^\star)$. Then it holds that $b \in P_k^{J^\star}$ for some $k \in \{1, \ldots, n\}$. Consider a source instance I' such that $A_k^{I'} = B_k^{I'} = \{b\}$, $R^{I'} = \emptyset$, and $A_i^{I'} = B_i^{I'} = \emptyset$ for every $i \in \{1, \ldots, n\}$ with $i \neq k$.

The target instance J' where $P_k^{J'} = \{b\}$ and $P_i^{J'} = \emptyset$ for every $i \in \{1, \ldots, n\}$ with $i \neq k$, is such that $(I', J') \in \mathcal{M}_n$. Notice that $J' \subseteq J^\star$. Now since Σ_n is a set of st-tgds, we know that $\mathcal{M}_n$ is closed-up on the right, obtaining that $(I', J^\star) \in \mathcal{M}_n$. Thus, given that $(J^\star, I_R) \in \mathcal{M}'$ we have that $(I', I_R) \in \mathcal{M}_n \circ \mathcal{M}'$. This last fact contradicts Proposition 3.8 since $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}_n$ and $\text{Sol}_{\mathcal{M}_n}(I_R) \not\subseteq \text{Sol}_{\mathcal{M}_n}(I')$.

We claim now that there must exist a dependency $\sigma \in \Sigma'$ such that $J^\star$ satisfies the premise of σ. Assume that this is not the case. Then since Σ' is a set of $\text{CQ}^{\mathbf{C}}$-TO-$\text{UCQ}^{=}$ formulas, it would be the case that $(J^\star, I_\emptyset) \models \Sigma'$, where $I_\emptyset$ is the empty source instance. Thus, we have that $(I_R, I_\emptyset) \in \mathcal{M}_n \circ \mathcal{M}'$ which, by Proposition 3.8, contradicts the fact that $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}_n$ since $\text{Sol}_{\mathcal{M}_n}(I_\emptyset) \not\subseteq \text{Sol}_{\mathcal{M}_n}(I_R)$. Assume now that σ is a dependency in Σ' whose premise is satisfied by $J^\star$. We show next that the premise and the conclusion of σ must be Boolean formulas. On the contrary, assume that σ is of the form $\varphi(\bar{x}) \rightarrow \psi(\bar{x})$, where $\bar{x}$ is a tuple of m variables with $m > 0$. Since we are assuming that $J^\star$ satisfies the premise of σ, there exists an m-tuple $\bar{b}$ such that $J^\star \models \varphi(\bar{b})$. We know that $\varphi(\bar{x})$ is a domain-independent formula, then it holds that every component of $\bar{b}$ is in $\text{dom}(J^\star)$. We have shown before that $\text{dom}(J^\star)$ is composed only by nulls and, thus, every component of $\bar{b}$ is a null value. Now, since $(J^\star, I_R) \models \sigma$ and $J^\star \models \varphi(\bar{b})$, it must be the case that $I_R \models \psi(\bar{b})$. We also know that $\psi(\bar{x})$ is domain independent, then every component of $\bar{b}$ must be in $\text{dom}(I_R)$, which leads to a contradiction since $\text{dom}(I_R) = \{a\}$ and $a \in \mathbf{C}$. In the rest of the proof, we let $\Sigma'' \subseteq \Sigma'$ be the set of all the dependencies σ of the form $\varphi \rightarrow \psi$ such that $J^\star \models \varphi$, where φ and ψ are Boolean formulas. Notice that $\Sigma'' \neq \emptyset$.

We have the necessary ingredients to show that Σ' is of size $\Omega(2^n)$. Consider for every n-tuple $\bar{d} = (d_1, \ldots, d_n) \in \{0, 1\}^n$, the set of source relation symbols $\mathbf{S}_{\bar{d}} = \{U_1(\cdot), \ldots, U_n(\cdot)\}$ such that $U_i = A_i$ if $d_i = 0$ and $U_i = B_i$ if $d_i = 1$. We now show that for each of the 2^n tuples $\bar{d}$, there must exist a dependency $\sigma \in \Sigma''$ of the form $\varphi \rightarrow \psi$ such that ψ has a disjunct that mentions exactly the relation symbols in $\mathbf{S}_{\bar{d}}$. This is enough to show that Σ' is of size $\Omega(2^n)$. Fix a tuple $\bar{d}$ and consider a source instance $I_{\bar{d}}$ such that for every $U \in \mathbf{S}_n$, if $U \in \mathbf{S}_{\bar{d}}$ then $U^{I_{\bar{d}}} = \{a\}$, otherwise $U^{I_{\bar{d}}} = \emptyset$. Since $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}_n$, there exists a target instance $J_{\bar{d}}$ such that $(I_{\bar{d}}, J_{\bar{d}}) \models \Sigma_n$ and $(J_{\bar{d}}, I_{\bar{d}}) \models \Sigma'$. Let J_P be a target instance such that $P_i^{J_P} = \{a\}$ for every $i \in \{1, \ldots, n\}$. It is straightforward to see that $J_P \subseteq J_{\bar{d}}$. It is also easy to see that, if θ is a Boolean query in $\text{CQ}^{\mathbf{C}}$ over $\mathbf{T}_n$, then $J_P \models \theta$. To see this just take a homomorphism h from the conjunctions of θ to the facts in J_P, such that $h(x) = a$ for every existential variable in θ, and note that $\mathbf{C}(h(x))$ holds for every variable, since $a \in \mathbf{C}$. Thus, given that queries in $\text{CQ}^{\mathbf{C}}$ are monotone and $J_P \subseteq J_{\bar{d}}$, we have that $J_{\bar{d}} \models \theta$ for every $\text{CQ}^{\mathbf{C}}$ Boolean query θ over $\mathbf{T}_n$. In particular, we have that for every $\varphi \rightarrow \psi \in \Sigma''$, it holds that $J_{\bar{d}} \models \varphi$. Then it must hold that $I_{\bar{d}} \models \psi$ for every $\varphi \rightarrow \psi \in \Sigma''$. This last fact implies that for every $\varphi \rightarrow \psi \in \Sigma''$, there exists a formula α such that α is one of the disjunctions of ψ and $I_{\bar{d}} \models \alpha$ (recall that ψ is a query in $\text{UCQ}^{=}$). Let Γ be a set containing all such formulas α, that is α is a formula in Γ if and only if there exists a dependency $\varphi \rightarrow \psi \in \Sigma''$ such that α is a disjunction in ψ

and $I_{\bar{d}} \models \alpha$. Note that every $\alpha \in \Gamma$ is a CQ$^{=}$ Boolean query, and since $I_{\bar{d}} \models \alpha$, it could not be the case that α mentions relation symbols of $\mathbf{S}_n$ outside $\mathbf{S}_{\bar{d}}$. We now show that one of the queries in Γ mentions exactly the relation symbols in $\mathbf{S}_{\bar{d}}$. On the contrary, assume that for every $\alpha \in \Gamma$, it is the case that α mentions a proper subset of the relation symbols of $\mathbf{S}_{\bar{d}}$. Consider for every $\alpha \in \Gamma$ a fresh constant value c_α, and a source instance I_α such that for every $U \in \mathbf{S}_n$, we have $U^{I_\alpha} = \{c_\alpha\}$ if the relation symbol U is mentioned in α, and $U^{I_\alpha} = \emptyset$ otherwise. It is clear that $I_\alpha \models \alpha$ for every $\alpha \in \Gamma$. Let $I_\Gamma = \bigcup_{\alpha \in \Gamma} I_\alpha$. Notice that for every $\alpha \in \Gamma$, it holds that $I_\Gamma \models \alpha$. Recall that for every $\varphi \rightarrow \psi \in \Sigma''$, there exists a formula $\alpha \in \Gamma$ such that α is one of the disjunctions of ψ. Hence, we obtain that $I_\Gamma \models \psi$ for every $\varphi \rightarrow \psi \in \Sigma''$. We also know that $J^\star \models \varphi$ for every $\varphi \rightarrow \psi \in \Sigma''$, obtaining that $(J^\star, I_\Gamma) \models \Sigma''$. Notice that Σ'' contains all the dependencies of Σ' such that $J^\star$ satisfies their premises and, thus, $(J^\star, I_\Gamma) \models \Sigma'$. Then since $(I_R, J^\star) \models \Sigma_n$ and $(J^\star, I_\Gamma) \models \Sigma'$, we have that $(I_R, I_\Gamma) \in \mathcal{M}_n \circ \mathcal{M}'$. We show now that $\mathrm{Sol}_{\mathcal{M}_n}(I_\Gamma) \not\subseteq \mathrm{Sol}_{\mathcal{M}_n}(I_R)$, which contradicts Proposition 3.8. Notice first that for every target instance $J \in \mathrm{Sol}_{\mathcal{M}_n}(I_R)$, there exists an element $c \in \mathrm{dom}(J)$ such that $c \in P_i^J$ for every $i \in \{1, \ldots, n\}$. We prove that there exists an instance in $\mathrm{Sol}_{\mathcal{M}_n}(I_\Gamma)$ that does not satisfy this last property. Consider for every $\alpha \in \Gamma$ the target instance $J_\alpha = \mathrm{chase}_{\Sigma_n}(I_\alpha)$, and let $J_\Gamma = \bigcup_{\alpha \in \Gamma} J_\alpha$. It is easy to see that $J_\Gamma \in \mathrm{Sol}_{\mathcal{M}_n}(I_\Gamma)$. Notice that since every $\alpha \in \Gamma$ mentions a proper subset of the relation symbols of $\mathbf{S}_{\bar{d}}$, there exists an index $i \in \{1, \ldots, n\}$ such that $A_i^{I_\alpha} = B_i^{I_\alpha} = \emptyset$, and then there exists an index $i \in \{1, \ldots, n\}$ such that $P_i^{J_\alpha} = \emptyset$. Moreover, since $\mathrm{dom}(J_\alpha) \cap \mathrm{dom}(J_{\alpha'}) = \emptyset$ for every pair of distinct elements α, α' of Γ, we obtain that there is no element $c \in \mathrm{dom}(J_\Gamma)$ such that $c \in P_i^{J_\Gamma}$ for every $i \in \{1, \ldots, n\}$. Thus, $J_\Gamma \notin \mathrm{Sol}_{\mathcal{M}_n}(I_R)$ implying that $\mathrm{Sol}_{\mathcal{M}_n}(I_\Gamma) \not\subseteq \mathrm{Sol}_{\mathcal{M}_n}(I_R)$, which leads to the contradiction previously mentioned. We have shown that there exists a formula $\alpha \in \Gamma$ such that α mentions exactly the relation symbols in $\mathbf{S}_{\bar{d}}$. Thus, there exists a dependency $\sigma \in \Sigma'' \subseteq \Sigma'$ such that the conclusion of σ has a disjunct that mentions exactly the relation symbols in $\mathbf{S}_{\bar{d}}$. This last property holds for every one of the 2^n distinct tuples $\bar{d}$, which implies that Σ' is of size exponential in the size of Σ_n. □

7.3 Computing Maximum Recoveries in the Full Case

Recall that a full FO-TO-CQ dependency does not include any existential quantifiers in its conclusion. In this section, we show that for mappings given by full FO-TO-CQ dependencies, maximum recoveries can be computed in polynomial time. This result is based on the fact that given a query composed of a single atom and with no existentially quantified variables, one can compute a rewriting of that query in quadratic time. This is formalized in the following lemma, where $\|\Sigma\|$ denotes the size of Σ.

LEMMA 7.6. *There exists an algorithm* QUERYREWRITINGATOM *that given an st-mapping* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$*, with* Σ *a set of* FO-TO-CQ *dependencies, and a conjunctive query* Q *over schema* $\mathbf{T}$ *composed by a single atom and with no existential quantifiers, computes in time* $O(\|\Sigma\|^2)$ *a domain-independent* FO *query* Q' *that is a rewriting of* Q *over the source. Moreover, if* Σ *is a set of full* FO-TO-CQ

st-dependencies where each dependency has a single atom in its conclusion, then the algorithm runs in time $O(\|\Sigma\|)$.

PROOF. The proof of the lemma is given in electronic Appendix A.2. □

By using algorithm QUERYREWRITINGATOM, we can compute in quadratic time a maximum recovery for mappings given by full dependencies.

Algorithm MAXIMUMRECOVERYFULL($\mathcal{M}$)

Input: An st-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where Σ is a set of full FO-TO-CQ dependencies, each dependency with a single atom in its conclusion.
Output: A ts-mapping $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$, where Σ' is a set of CQ-TO-FO dependencies and $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$.

(1) Start with Σ' as the empty set.
(2) For every atom $R(\bar{x})$ that is the conclusion of a dependency in Σ, do the following:
  (a) Let Q be the conjunctive query defined by $R(\bar{x})$.
  (b) Use QUERYREWRITINGATOM($\mathcal{M}$, Q) to compute an FO formula $\alpha(\bar{x})$ that is a rewriting of $R(\bar{x})$ over the source.
  (c) Add dependency $R(\bar{x}) \rightarrow \alpha(\bar{x})$ to Σ'.
(3) Return $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$.

THEOREM 7.7. *Let* $\mathcal{M}$ *be an st-mapping specified by a set* Σ *of full* FO-TO-CQ *st-dependencies, each dependency with a single atom in its conclusion. Then* MAXIMUMRECOVERYFULL($\mathcal{M}$) *computes a maximum recovery of* $\mathcal{M}$ *in time* $O(\|\Sigma\|^2)$*, which is specified by a set of* CQ-TO-FO *dependencies.*

PROOF. Since Σ is a set of full FO-TO-CQ st-dependencies, each dependency with a single atom in its conclusion, algorithm QUERYREWRITINGATOM($\mathcal{M}$, Q) runs in linear time. Thus, it is straightforward to see that algorithm MAXIMUMRECOVERYFULL runs in quadratic time. The correctness of the algorithm follows from the proof of Theorem 7.3. We only notice here that the output of algorithm MAXIMUMRECOVERYFULL does not include predicate $\mathbf{C}(\cdot)$. Since Σ is a set of full dependencies, $\text{chase}_\Sigma(I)$ is composed only by constant values and, thus, $\mathbf{C}(\cdot)$ is not needed in the proof of Theorem 7.3. □

Notice that in Theorem 7.7, we assume that every dependency has a single atom in its conclusion. Nevertheless, this theorem can be extended to the general case; from a set Σ of arbitrary full FO-TO-CQ st-dependencies, one can obtain as follows, an equivalent set Σ' of full FO-TO-CQ st-dependencies having a single atom in the conclusion of each constraint. For every dependency $\varphi(\bar{x}) \rightarrow \psi(\bar{x})$ in Σ and atom $R(\bar{y})$ in $\psi(\bar{x})$, where $\bar{y} \subseteq \bar{x}$, the dependency $\varphi(\bar{x}) \rightarrow R(\bar{y})$ is included in Σ'. Thus, to apply Theorem 7.7 to Σ, we first construct Σ' from Σ and then apply procedure MAXIMUMRECOVERYFULL. It is important to notice that Σ' could be of quadratic size in the size of Σ, and hence, by the fact that algorithm QUERYREWRITINGATOM runs in linear time and the definition of procedure MAXIMUMRECOVERYFULL, it follows that a maximum recovery for a mapping specified by an arbitrary set of full FO-TO-CQ st-dependencies can be computed in cubic-time.

As for the general case, from Theorems 6.3 and 6.4, we know that this algorithm computes an inverse (quasi-inverse) if Σ is an invertible (quasi-invertible) set of full st-tgds. The algorithm in Fagin et al. [2008] for computing an inverse of a set Σ of full st-tgds returns a set Σ' of CQ$^{\neq}$-TO-CQ dependencies of exponential size in $\|\Sigma\|$. The algorithm in Fagin et al. [2008] for computing a quasi-inverse of a set Σ of full st-tgds returns a set Σ' of CQ$^{\neq}$-TO-UCQ dependencies, which is also of exponential size in $\|\Sigma\|$. In both cases, our algorithm works in quadratic time and returns a set Σ' of CQ-TO-UCQ$^{=}$ dependencies, which is of quadratic size in $\|\Sigma\|$.

7.4 Computing Maximum Recoveries for Mappings with Source Dependencies

We conclude this section by showing how algorithm MAXIMUMRECOVERY can be extended to handle arbitrary source constraints.

By using Lemma 4.4, we can extend algorithm MAXIMUMRECOVERY to handle source constraints. Given an st-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{\mathbf{st}}, \Gamma_{\mathbf{s}})$, where $\Sigma_{\mathbf{st}}$ is a set of FO-TO-CQ st-dependencies (from $\mathbf{S}$ to $\mathbf{T}$) and $\Gamma_{\mathbf{s}}$ is a set of source FO-dependencies (over $\mathbf{S}$), algorithm MAXIMUMRECOVERY can be used to produce a maximum recovery $\mathcal{M}_1^{\star} = (\mathbf{T}, \mathbf{S}, \Sigma_{\mathbf{ts}})$ for st-mapping $\mathcal{M}_1 = (\mathbf{S}, \mathbf{T}, \Sigma_{\mathbf{st}})$, where $\Sigma_{\mathbf{ts}}$ is a set of CQ$^{\mathbf{C}}$-TO-FO ts-dependencies from $\mathbf{T}$ to $\mathbf{S}$, and then $\mathcal{M}^{\star} = (\mathbf{T}, \mathbf{S}, \Sigma_{\mathbf{ts}}, \Gamma_{\mathbf{s}})$ is output as a maximum recovery of $\mathcal{M}$.

8. THE LANGUAGE OF MAXIMUM RECOVERIES

Given a mapping $\mathcal{M}$ specified by a set of FO-TO-CQ dependencies, algorithm MAXIMUMRECOVERY produces a maximum recovery of $\mathcal{M}$ that is specified by a set of CQ$^{\mathbf{C}}$-TO-FO dependencies. In this section, we study some properties of the language needed to express maximum recoveries, which provides justification for the language used in the output of algorithm MAXIMUMRECOVERY. Moreover, we also show that the extension of this algorithm to handle target constraints is not immediate, as there exists a mapping specified by a set of FO-TO-CQ dependencies plus a set of target egds that has no maximum recovery specified by a set of FO-sentences, and the same holds for a weakly acyclic set of target tgds.

A first question about the output of MAXIMUMRECOVERY is whether predicate $\mathbf{C}(\cdot)$ is really needed. In Fagin et al. [2008], it is proved that $\mathbf{C}(\cdot)$ is needed when computing quasi-inverses of st-mappings specified by st-tgds, if quasi-inverses are expressed using st-tgds with inequalities in the premises and disjunction in the conclusions. Here we show that $\mathbf{C}(\cdot)$ is needed when computing maximum recoveries for st-mappings specified by st-tgds, even if we allow the full power of FO to express maximum recoveries.

PROPOSITION 8.1. *There exists an st-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ specified by a set Σ of st-tgds that has no maximum recovery specified by a set of* FO*-sentences over $\mathbf{S} \cup \mathbf{T}$ not using predicate $\mathbf{C}(\cdot)$.*

PROOF. The proof of the proposition is given in electronic Appendix B.1. □

In Section 4, it is proved that adding disjunction, inequalities or negation to the conclusions of FO-TO-CQ dependencies generates st-mappings that do not

necessarily have maximum recoveries. Hence, it would be desirable to stay in the class of FO-TO-CQ dependencies when dealing with maximum recoveries. In particular, it would be desirable to have an algorithm that takes as input a set Σ of FO-TO-CQ st-dependencies, and produces a set Σ' of $\text{FO}^{\mathbf{C}}$-TO-CQ ts-dependencies which is a maximum recovery of Σ. Thus, a second important question about the algorithm MAXIMUMRECOVERY is whether it could be modified to produce a set of $\text{FO}^{\mathbf{C}}$-TO-CQ ts-dependencies as output. Unfortunately, the following proposition shows that this could not be the case, even if we allow disjunction in the conclusions of the output dependencies.

PROPOSITION 8.2. *There exists an st-mapping specified by a set of* FO-TO-CQ *st-dependencies that has no maximum recovery specified by a set of* $\text{FO}^{\mathbf{C}}$-TO-UCQ *ts-dependencies.*

PROOF. The proof of the proposition is given in electronic Appendix B.2. □

From the proof of Proposition 8.2 , we obtain that there exists an st-mapping specified by a set of $\text{CQ}^{\neq}$-TO-CQ dependencies that has no maximum recovery specified by a set of $\text{FO}^{\mathbf{C}}$-TO-UCQ dependencies.

A third question about the output of MAXIMUMRECOVERY is whether the full power of FO is really needed in the conclusions of the dependencies returned by the algorithm. For example, could it be the case that $\text{CQ}^{\mathbf{C}}$-TO-$\text{UCQ}^{=,\neg}$ dependencies suffice to specify maximum recoveries for st-mappings specified by FO-TO-CQ dependencies? Theorem 8.3 shows that this could not be the case. In fact, we show that for $\mathcal{L}$ and $\mathcal{L}'$ fragments of FO (satisfying some regularity conditions), if $\text{CQ}^{\mathbf{C}}$-TO-$\mathcal{L}'$ dependencies suffice to specify maximum recoveries for mappings given by $\mathcal{L}$-TO-CQ dependencies, then $\mathcal{L}'$ must be at least as expressive as $\mathcal{L}$.

In Theorem 8.3, we use the following terminology. We say that a fragment $\mathcal{L}$ of FO is closed under conjunction and existential quantification, if for every pair of formulas φ and ψ in $\mathcal{L}$, there exist formulas α and β in $\mathcal{L}$ such that, α is equivalent to $\varphi \wedge \psi$ and β is equivalent to $\exists x\, \varphi$. Furthermore, we say that $\mathcal{L}$ is closed under free-variable substitution, if for every formula $\varphi(\bar{x})$ in $\mathcal{L}$ and substitution μ for $\bar{x}$, there exists a formula $\alpha(\mu(\bar{x}))$ in $\mathcal{L}$ that is equivalent to $\varphi(\mu(\bar{x}))$. Notice that all the fragments of FO used in this article are closed under conjunction, existential quantification and free-variable substitution. Finally, we say that an FO-sentence Φ is nontrivial if Φ is neither a contradiction nor a valid sentence.

THEOREM 8.3. *Let $\mathcal{L}$ and $\mathcal{L}'$ be fragments of* FO *that are closed under conjunction, existential quantification, and free-variable substitution. If there exists a nontrivial sentence Φ in $\mathcal{L}$ that is not equivalent to any sentence in $\mathcal{L}'$, then there exists an st-mapping specified by a set of $\mathcal{L}$-TO-CQ st-dependencies that has no maximum recovery specified by a set of* $\text{CQ}^{\mathbf{C}}$-TO-$\mathcal{L}'$ *ts-dependencies.*

PROOF. The proof of the theorem is given in electronic Appendix B.3. □

In Section 7, we show that algorithm MAXIMUMRECOVERY can be extended to handle arbitrary source constraints. In Theorem 4.7, we show that if an

st-mapping $\mathcal{M}$ is specified by a set of FO-TO-CQ dependencies, a set of target egds and a weakly acyclic set of target tgds, then $\mathcal{M}$ has a maximum recovery. Thus, a natural question is whether MAXIMUMRECOVERY can be extended to this class of mappings with target dependencies. Unfortunately, the following proposition shows that the extension of the algorithm to handle target constraints is by no means immediate.

PROPOSITION 8.4.

(1) *There exists an st-mapping $\mathcal{M}$ specified by a set of st-tgds plus a set of target egds that has no maximum recovery specified by a set of* FO*-sentences.*
(2) *There exists an st-mapping $\mathcal{M}$ specified by a set of st-tgds plus a weakly acyclic set of target tgds that has no maximum recovery specified by a set of* FO*-sentences.*

PROOF. The proof of the proposition is given in electronic Appendix B.4. □

9. COMPLEXITY RESULTS

In Fagin [2007], two problems are identified as important decision problems for the notion of inverse: (1) to check whether a mapping $\mathcal{M}$ is invertible, and (2) to check whether a mapping $\mathcal{M}_2$ is an inverse of a mapping $\mathcal{M}_1$. These questions are considered in the context of st-tgds in Fagin [2007], where they are also relevant for the notion of quasi-inverse [Fagin et al. 2008]. In this context, the problem of verifying whether a mapping $\mathcal{M}$ has a maximum recovery becomes trivial, as every mapping specified by this type of dependency admits a maximum recovery. In fact, this question is also trivial for the larger class of mappings specified by FO-TO-CQ dependencies. The goal of this section is to show that the problem of verifying, given mappings $\mathcal{M}$ and $\mathcal{M}'$, whether $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$, is undecidable. To this end, we prove a stronger result, namely that undecidability still holds if maximum recovery is replaced by the weaker notion of recovery in the previous problem. We start by considering mappings specified by full st-tgds.

PROPOSITION 9.1. *The problem of verifying, given mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$, where Σ is a set of full st-tgds and Σ' is a set of ts-tgds, whether $\mathcal{M}'$ is a recovery of $\mathcal{M}$, is Π_2^P-complete. Moreover, if Σ' is a set of full ts-tgds, then this problem is coNP-complete.*

We note that the problem considered in this proposition becomes undecidable if Σ is a set of full FO-TO-CQ dependencies (this is a straightforward consequence of the undecidability of the problem of verifying whether an FO sentence is finitely satisfiable [Libkin 2004]). For this reason, in this section we focus on st-tgds. To prove the proposition, we start by showing a simple but useful lemma.

LEMMA 9.2. *Let $\mathcal{M}$ be an st-mapping specified by a set of st-tgds. Assume that $\mathcal{M}'$ is a ts-mapping such that whenever $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, it holds that $I_1 \subseteq I_2$. Then $\mathcal{M}'$ is an inverse of $\mathcal{M}$ if and only if $\mathcal{M}'$ is a recovery of $\mathcal{M}$.*

PROOF. The proof of the lemma is given in electronic Appendix C.1. □

To prove Proposition 9.1, we also need to introduce some terminology and prove a technical lemma. Let Σ be a set of CQ-TO-CQ dependencies from a schema $\mathbf{R}_1$ to a schema $\mathbf{R}_2$ and I an instance of $\mathbf{R}_1$. We denote by k_Σ the maximum, over all members $\varphi \in \Sigma$, of the number of conjuncts that appear in the premise of φ, and by $|I|$ the total number of tuples in I, that is, $|I| = \sum_{R \in \mathbf{R}_1} |R^I|$, where $|R^I|$ is the number of tuples in R^I. Moreover, we define the notion of I being $\mathbf{N}$-*connected* as follows. Let $G_I = (V_I, E_I)$ be a graph such that: (1) V_I is the set of all tuples $t \in R^I$, for some $R \in \mathbf{R}_1$, and (2) a tuple $(t_1, t_2) \in E_I$ if and only if there exists a null value $n \in \mathbf{N}$ that is mentioned both in t_1 and t_2. Then I is $\mathbf{N}$-connected if the graph G_I is connected. An instance I_1 is an $\mathbf{N}$-*connected sub-instance* of I, if I_1 is a sub-instance of I and I_1 is $\mathbf{N}$-connected. Finally, I_1 is an $\mathbf{N}$-*connected component* of I, if I_1 is an $\mathbf{N}$-connected sub-instance of I and there is no $\mathbf{N}$-connected sub-instance I_2 of I such that I_1 is a proper sub-instance of I_2.

LEMMA 9.3. *Let* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ *and* $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$ *be schema mappings, where* Σ *is a set of full st-tgds and* Σ' *is a set of ts-tgds. Then* $\mathcal{M}'$ *is a recovery of* $\mathcal{M}$ *if and only if for every source instance* I *such that* $|I| \leq k_\Sigma \cdot k_{\Sigma'}$ *and* $\mathbf{N}$*-connected component* K *of* $\text{chase}_{\Sigma'}(\text{chase}_\Sigma(I))$*, there exists a homomorphism from* K *to* I *that is the identity on* $\mathbf{C}$.

PROOF. The proof of the lemma is given in electronic Appendix C.2. □

PROOF OF PROPOSITION 9.1. First, we assume that Σ' is a set of full ts-tgds, and we show that the problem of verifying whether $\mathcal{M}'$ is not a recovery of $\mathcal{M}$ is NP-complete. From Lemma 9.3 and the fact that Σ' is a set of full ts-tgds, we have that $\mathcal{M}'$ is not a recovery of $\mathcal{M}$ if and only if there exists a source instance I such that $|I| \leq k_\Sigma \cdot k_{\Sigma'}$ and there exists a tuple in $\text{chase}_{\Sigma'}(\text{chase}_\Sigma(I))$ that is not in I. The latter is an NP property; to check whether it holds, it is enough to guess an instance I such that $|I| \leq k_\Sigma \cdot k_{\Sigma'}$, and then guess the chase steps that produce a tuple that is not in I. Thus, we have that the problem of verifying whether $\mathcal{M}'$ is not a recovery of $\mathcal{M}$ is in NP.

To show that the problem is coNP-hard we use a result from Fagin [2007]. In the proof of Theorem 14.9 in Fagin [2007], it was shown that, given a propositional formula φ, one can construct two mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$ with Σ and Σ' sets of full st-tgds and full ts-tgds, respectively, such that $\mathcal{M}'$ is an inverse of $\mathcal{M}$ if and only if φ is not satisfiable. Moreover, the mappings constructed in that proof were such that if $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$ then $I_1 \subseteq I_2$. Then from Lemma 9.2, we know that $\mathcal{M}'$ is an inverse of $\mathcal{M}$ if and only if $\mathcal{M}'$ is a recovery of $\mathcal{M}$. We have that $\mathcal{M}'$ is a recovery of $\mathcal{M}$ if and only if φ is not satisfiable. Thus, the hardness results follows then from the well-known fact that, testing whether a propositional formula is satisfiable is an NP-complete problem.

Second, we assume that Σ' is a set of ts-tgds, and we show that the problem of verifying whether $\mathcal{M}'$ is a recovery of $\mathcal{M}$ is Π_2^P-complete. From Lemma 9.3, we have that $\mathcal{M}'$ is a recovery of $\mathcal{M}$ if and only if for every source instance I such that $|I| \leq k_\Sigma \cdot k_{\Sigma'}$ and $\mathbf{N}$-connected component K of $\text{chase}_{\Sigma'}(\text{chase}_\Sigma(I))$, there exists a homomorphism from K to I that is the identity on $\mathbf{C}$. Given that

the size of I, as well as the size of K, is polynomial in the size of $\mathcal{M}$ and $\mathcal{M}'$, and that the homomorphism problem is in NP, we have that the problem of verifying whether $\mathcal{M}'$ is a recovery of $\mathcal{M}$ is in Π_2^P. To prove that this problem is indeed Π_2^P-complete, we give a reduction from the problem of verifying whether a quantified propositional formula:

$$\varphi = \forall u_1 \cdots \forall u_\ell \exists v_1 \cdots \exists v_m \, \psi, \tag{9}$$

is valid, where ψ is a 3-CNF propositional formula. This problem is known to be Π_2^P-complete [Du and Ko 2000].

Let $\mathbf{S} = \{TV(\cdot,\cdot), R_0(\cdot,\cdot,\cdot), R_1(\cdot,\cdot,\cdot), R_2(\cdot,\cdot,\cdot), R_3(\cdot,\cdot,\cdot)\}$ and $\mathbf{T} = \{U_1(\cdot,\cdot,\cdot), \ldots, U_\ell(\cdot,\cdot,\cdot)\}$. Next we define schema mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$ such that, φ is valid if and only if $\mathcal{M}'$ is a recovery of $\mathcal{M}$. The first argument of predicate TV is used to store the truth value *true*, while its second argument is used to store the truth value *false*. Predicate R_0 is used to store the truth assignments that satisfy the clauses of the form $u \vee v \vee w$ (clauses without negative literals). Assuming that variables x, y store values *true* and *false*, respectively, the following formula is used to define R_0:

$$\varphi_0(x, y) = R_0(x,x,x) \wedge R_0(x,x,y) \wedge R_0(x,y,x) \wedge R_0(y,x,x) \wedge R_0(x,y,y) \wedge R_0(y,x,y) \wedge R_0(y,y,x).$$

Similarly, predicate R_1 is used to store the truth assignments that satisfy the clauses of the form $u \vee v \vee \neg w$, predicate R_2 is used to store the truth assignments that satisfy the clauses of the form $u \vee \neg v \vee \neg w$, and predicate R_3 is used to store the truth assignments that satisfy the clauses of the form $\neg u \vee \neg v \vee \neg w$. Again assuming that variables x, y store values *true* and *false*, respectively, the following formulas are used to define R_1, R_2 and R_3:

$$\begin{aligned}
\varphi_1(x, y) &= R_1(x,x,x) \wedge R_1(x,x,y) \wedge R_1(x,y,x) \wedge R_1(y,x,x) \wedge R_1(x,y,y) \wedge R_1(y,x,y) \wedge R_1(y,y,y),\\
\varphi_2(x, y) &= R_2(x,x,x) \wedge R_2(x,x,y) \wedge R_2(x,y,x) \wedge R_2(x,y,y) \wedge R_2(y,x,y) \wedge R_2(y,y,x) \wedge R_2(y,y,y),\\
\varphi_3(x, y) &= R_3(x,x,y) \wedge R_3(x,y,x) \wedge R_3(y,x,x) \wedge R_3(x,y,y) \wedge R_3(y,x,y) \wedge R_3(y,y,x) \wedge R_3(y,y,y).
\end{aligned}$$

Finally, the first argument of predicate U_i is used to store the truth value of propositional variable u_i, for every $i \in \{1, \ldots, \ell\}$. We include two extra arguments in U_i for a technical reason that will become clear when we prove that the reduction is correct.

Set Σ of full st-tgds is given by the following dependency:

$$T(x, y) \wedge \varphi_0(x, y) \wedge \varphi_1(x, y) \wedge \varphi_2(x, y) \wedge \varphi_3(x, y) \rightarrow U_1(x,x,y) \wedge U_1(y,x,y) \wedge \cdots \wedge U_\ell(x,x,y) \wedge U_\ell(y,x,y). \tag{10}$$

Set Σ' of ts-tgds is given by the following dependency:

$$U_1(u_1, x, y) \wedge \cdots \wedge U_\ell(u_\ell, x, y) \rightarrow \exists v_1 \cdots \exists v_m \, \theta(u_1, \ldots, u_\ell, v_1, \ldots, v_m), \tag{11}$$

where $\theta(u_1, \ldots, u_\ell, v_1, \ldots, v_m)$ is defined as follows. If 3-CNF formula ψ in (9) is equal to $C_1 \wedge \cdots \wedge C_k$, where each C_i is a clause, then $\theta = \theta_1 \wedge \cdots \wedge \theta_k$, where θ_i is obtained from C_i as follows. Without loss of generality, we assume that in C_i, the positive literals appear before the negative literals (if C_i has at least one positive literal). Then if $C_i = u \vee v \vee w$, we have that $\theta_i = R_0(u, v, w)$, if $C_i = u \vee v \vee \neg w$, we have that $\theta_i = R_1(u, v, w)$, if $C_i = u \vee \neg v \vee \neg w$, we have that $\theta_i = R_2(u, v, w)$, and if $C_i = \neg u \vee \neg v \vee \neg w$, we have that $\theta_i = R_3(u, v, w)$. For example, if $\varphi = \forall u_1 \forall u_2 \exists v_1 ((u_1 \vee v_1 \vee \neg u_2) \wedge (u_1 \vee u_2 \vee v_1))$, then:

$$\Sigma = \{TV(x, y) \wedge \varphi_0(x, y) \wedge \varphi_1(x, y) \wedge \varphi_2(x, y) \wedge \varphi_3(x, y) \rightarrow U_1(x, x, y) \wedge U_1(y, x, y) \wedge U_2(x, x, y) \wedge U_2(y, x, y)\},$$

$$\Sigma' = \{U_1(u_1, x, y) \wedge U_2(u_2, x, y) \rightarrow \exists v_1 (R_1(u_1, v_1, u_2) \wedge R_0(u_1, u_2, v_1))\}.$$

Next we show that φ is valid if and only if $\mathcal{M}'$ is a recovery of $\mathcal{M}$.

($\Rightarrow$) Assume that φ is valid. From Lemma 9.3, to show that $\mathcal{M}'$ is a recovery of $\mathcal{M}$, it is enough to prove that for every instance I of $\mathbf{S}$ and $\mathbf{N}$-connected component K of $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I))$, there exists a homomorphism from K to I that is the identity on $\mathbf{C}$. Next we show that this is the case.
Let I_1 be an instance of $\mathbf{S}$. By definition of Σ and Σ', and in particular because of the inclusion of the two extra arguments in each U_i, we have that if K is an $\mathbf{N}$-connected component of $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I_1))$, then there exists a pair of values a, b in $\text{dom}(I_1)$ such that: (1) $I_1 \models T(a, b) \wedge \varphi_0(a, b) \wedge \varphi_1(a, b) \wedge \varphi_2(a, b) \wedge \varphi_3(a, b)$, (2) $\text{chase}_{\Sigma}(I_1) \models U_1(c_1, a, b) \wedge \cdots \wedge U_\ell(c_\ell, a, b)$, where each c_i is either a or b, and (3) K is generated from $\exists v_1 \cdots \exists v_m \, \theta(c_1, \ldots, c_\ell, v_1, \ldots, v_m)$ when computing $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I_1))$. Assume that in the construction of K, variable v_i is replaced by value $n_i \in \mathbf{N}$, for every $i \in \{1, \ldots, m\}$. Given that φ is valid, we know that for the truth assignment σ_1 such that $\sigma_1(u_i) = c_i$, for every $i \in \{1, \ldots, \ell\}$, there exists a truth assignment σ_2 such that $\sigma_1 \cup \sigma_2$ satisfies propositional formula ψ in (9). From this we conclude that function h defined as $h(n_i) = \sigma_2(v_i)$ ($i \in \{1, \ldots, m\}$) and $h(c) = c$ ($c \in \mathbf{C}$) is a homomorphism from K to I that is the identity on $\mathbf{C}$.

($\Leftarrow$) Assume that $\mathcal{M}'$ is a recovery of $\mathcal{M}$, and let I be an instance of $\mathbf{S}$ such that $T^I = \{(a, b)\}$, where a and b are two distinct elements from $\mathbf{C}$, and

$$\begin{aligned}
R_0^I &= \{(a,a,a), (a,a,b), (a,b,a), (b,a,a), (a,b,b), (b,a,b), (b,b,a)\},\\
R_1^I &= \{(a,a,a), (a,a,b), (a,b,a), (b,a,a), (a,b,b), (b,a,b), (b,b,b)\},\\
R_2^I &= \{(a,a,a), (a,a,b), (a,b,a), (a,b,b), (b,a,b), (b,b,a), (b,b,b)\},\\
R_3^I &= \{(a,a,b), (a,b,a), (b,a,a), (a,b,b), (b,a,b), (b,b,a), (b,b,b)\}.
\end{aligned}$$

Given that $\mathcal{M}'$ is a recovery of $\mathcal{M}$, we have that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$. Thus, for every tuple $(c_1, \ldots, c_\ell) \in \{a, b\}^\ell$, there exists a tuple $(d_1, \ldots, d_m) \in \{a, b\}^m$ such that $I \models \theta(c_1, \ldots, c_\ell, d_1, \ldots, d_m)$. Hence, by the definitions of θ, R_0^I, R_1^I, R_2^I and R_3^I, we conclude that φ is a valid formula. This concludes the proof of the theorem. □

Proposition 9.1 is in sharp contrast with the results of Fagin [2007], where it is shown that the problem of verifying, given schema mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$, with Σ a set of full st-tgds and Σ' a set of full ts-tgds, whether $\mathcal{M}'$ is an inverse of $\mathcal{M}$ is DP-complete.[1] The lower complexity for the case of the recovery is not surprising, as the notion of recovery is much weaker than the notion of inverse. However, the situation is different for the case of non-full st-tgds.

THEOREM 9.4. *The problem of verifying, given mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$, where Σ is a set of st-tgds and Σ' is a set of ts-tgds, whether $\mathcal{M}'$ is a recovery of $\mathcal{M}$ is undecidable.*

PROOF. The proof of the theorem is given in electronic Appendix C.3. □

As a corollary of Theorem 9.4 and the results in Section 6, we obtain the following undecidability results for maximum recoveries, inverses[2] and quasi-inverses.

COROLLARY 9.5. *The problems of verifying, given mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$, where Σ is a set of st-tgds and Σ' is a set of ts-tgds, whether (1) $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$, (2) $\mathcal{M}'$ is an inverse of $\mathcal{M}$, and (3) $\mathcal{M}'$ is a quasi-inverse of $\mathcal{M}$, are all undecidable.*

PROOF. The proof of the corollary is given in electronic Appendix C.4. □

10. CONCLUDING REMARKS

In this article, we introduce the notion of a recovery of a mapping: a reverse mapping that recovers sound information. We introduce an order relation on recoveries, from which the notion of maximum recovery naturally arises. As our results show, maximum recoveries possess good properties that justify their usage in data exchange and metadata management. Most notably, maximum recoveries exist for the large class of mappings specified by FO-TO-CQ dependencies.

An important open problem is the decidability of the existence of maximum recoveries for classes of dependencies beyond FO-TO-CQ, for example the classes of FO-TO-UCQ and FO-TO-$\text{CQ}^{\neq}$ dependencies. Although we have concentrated on the relational case, a characteristic of the notions of recovery and maximum recovery is that they are bounded neither to a specific data model nor to a specific language for expressing schema mappings. As part of our future work, we plan to study these notions for other semantics, for example, closed world semantics [Libkin 2006], and for other data models, for example, XML.

[1] A problem is in DP if it is the intersection of an NP problem and a coNP problem [Papadimitriou 1993].

[2] The undecidability of the problem of verifying whether a mapping $\mathcal{M}'$ is an inverse of a mapping $\mathcal{M}$ was already mentioned in Fagin [2007]. As pointed out in that paper, this result was actually proved by the first author (M. Arenas) in an unpublished manuscript.

ACKNOWLEDGMENTS

We are very grateful to Pablo Barceló, Leopoldo Bertossi, Alejandro Cataldo, Balder ten Cate, Leonid Libkin, and Juan Reutter, for many helpful discussions, and to the anonymous referees for their comments.

REFERENCES

Abiteboul, S. and Duschka, O. M. 1998. Complexity of answering queries using materialized views. In *Proceedings of the 17th ACM Symposium on Principles of Database Systems (PODS)*, 254–263.

Afrati, F. N., Li, C., and Pavlaki, V. 2008. Data exchange in the presence of arithmetic comparisons. In *Proceedings of the 11th International Conference on Extending Database Technology (EDBT)*, 487–498.

Arenas, M., Barceló, P., Fagin, R., and Libkin, L. 2004. Locally consistent transformations and query answering in data exchange. In *Proceedings of the 23rd ACM Symposium on Principles of Database Systems (PODS)*, 229–240.

Arenas, M., Pérez, J., and Riveros, C. 2008. The recovery of a schema mapping: Bringing exchanged data back. In *Proceedings of the 28th ACM Symposium on Principles of Database Systems (PODS)*, 13–22.

Bernstein, P. 2003. Applying model management to classical meta data problems. In *Proceedings of the 1st Biennial Conference on Innovative Data Systems Research (CIDR)*.

Bernstein, P. and Melnik, S. 2007. Model management 2.0: Manipulating richer mappings. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 1–12.

De Giacomo, G., Lembo, D., Lenzerini, M., and Rosati, R. 2007. On reconciling data exchange, data integration, and peer data management. In *Proceedings of the 26th ACM Symposium on Principles of Database Systems (PODS)*, 133–142.

Deutsch, A. and Tannen, V. 2003. Reformulation of XML queries and constraints. In *Proceedings of the 9th International Conference on Database Theory (ICDT)*, 225–241.

Du, D.-Z. and Ko, K.-I. 2000. *Theory of Computational Complexity*. Wiley-Interscience.

Duschka, O. M. and Genesereth, M. R. 1997. Answering recursive queries using views. In *Proceedings of the 16th ACM Symposium on Principles of Database Systems (PODS)*, 109–116.

Fagin, R. 1982. Horn clauses and database dependencies. *J. ACM. 29*, 4, 952–985.

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

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

Fagin, R., Kolaitis, P. G., and Popa, L. 2005b. Data exchange: Getting to the core. *ACM Trans. Datab. Syst. 30*, 1, 174–210.

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

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

Gottlob, G. and Nash, A. 2006. Data exchange: Computing cores in polynomial time. In *Proceedings of the 25th ACM Symposium on Principles of Database Systems (PODS)*, 40–49.

Halevy, A. 2000. Theory of answering queries using views. *SIGMOD Record 29*, 1, 40–47.

Halevy, A. 2001. Answering queries using views: A survey. *VLDB J. 10*, 4, 270–294.

Hernich, A. and Schweikardt, N. 2009. Logic and data exchange: Which solutions are "good" solutions? Logic and the foundations of game and decision theory (LOFT 8). G. Bonanno, B. Löwe, and W. van der Hoek, *29*, 1, *Texts in Logic and Games*, Amsterdam University Press. To appear.

Kolaitis, P. G. 2005. Schema mappings, data exchange, and metadata management. In *Proceedings of the 24th ACM Symposium on Principles of Database Systems (PODS)*, 61–75.

Kolaitis, P. G., Panttaja, J., and Tan, W.-C. 2006. The complexity of data exchange. In *Proceedings of the 25th ACM Symposium on Principles of Database Systems (PODS)*, 30–39.

LENZERINI, M. 2002. Data integration: A theoretical perspective. In *Proceedings of the 21st ACM Symposium on Principles of Database Systems (PODS)*, 233–246.

LEVY, A. Y., MENDELZON, A. O., SAGIV, Y., AND SRIVASTAVA, D. 1995. Answering queries using views. In *Proceedings of the 14th ACM Symposium on Principles of Database Systems (PODS)*, 95–104.

LEVY, A. Y., RAJARAMAN A., AND ORDILLE, J. J. 1996. Querying heterogeneous information sources using source descriptions. In *Proceedings of the 22th International Conference on Very Large Data Bases (VLDB)*, 251–262.

LIBKIN, L. 2004. *Elements of Finite Model Theory*, 1st edition. Springer-Verlag.

LIBKIN, L. 2006. Data exchange and incomplete information. In *Proceedings of the 25th ACM Symposium on Principles of Database Systems (PODS)*, 60–69.

MELNIK, S. 2004. Generic model management: Concepts and algorithms. Lecture Notes in Computer Science, vol. 2967. Springer.

MELNIK, S., BERNSTEIN, P. A., HALEVY, A. Y., AND RAHM, E. 2005. Supporting executable mappings in model management. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 167–178.

PAPADIMITRIOU, C. H. 1993. *Computational Complexity*. Addison Wesley.

POTTINGER, R. AND HALEVY, A. Y. 2001. MiniCon: A scalable algorithm for answering queries using views. *VLDB J. 10*, 2–3, 182–198.

Received December 2008; revised July 2009; accepted August 2009