

The Recovery of a Schema Mapping: Bringing Exchanged Data Back

MARCELO ARENAS and JORGE PÉREZ

Pontificia Universidad Católica de Chile

and

CRISTIAN RIVEROS

R&M Tech Ingeniería y Servicios Limitada

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 paper, 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 [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

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 the last years, a lot of attention has been paid to the development of solid foundations for the problem of exchanging data using schema map-

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, Santiago, Chile; email: {marenas,jperez}@ing.puc.cl; C. Riveros, R&M Tech Ingeniería y Servicios Limitada, Santiago, Chile; email: criveros@rmt.cl.

Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.

© 20YY ACM 0362-5915/20YY/0300-0001 \$5.00

pings [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 [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 in [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]. In [Fagin 2007], Fagin proposes 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 introduces in [Fagin 2007] 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], Fagin et al. 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 non-invertible 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 $\overline{\text{Id}}$, that 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 paper, 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 that drive to

work. Assume that the information about employees has to be transferred to an independent database that contains relation $Shuttle(name)$, that stores the names of employees that take a shuttle bus to go to work. A schema mapping \mathcal{M}_{E-S} between these two databases is defined by the following dependency:

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

An example of a reverse mapping \mathcal{M}_1 that recovers sound information w.r.t. \mathcal{M}_{E-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 w.r.t. \mathcal{M}_{E-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 w.r.t. \mathcal{M}_{E-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-S} , while mapping \mathcal{M}_3 is not a recovery of \mathcal{M}_{E-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 that 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 w.r.t. \mathcal{M} . Furthermore, if \mathcal{M}_4 is a mapping defined by dependency:

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

then \mathcal{M}_4 is a recovery of \mathcal{M}_{E-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}'' w.r.t. \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-S} .

In this paper, 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 paper, 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 $\forall \bar{x}(\varphi_S(\bar{x}) \rightarrow \exists \bar{y} \psi_T(\bar{x}, \bar{y}))$, where $\varphi_S(\bar{x})$ is a first-order formula over the source schema and $\psi_T(\bar{x}, \bar{y})$ 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 between the notions of inverse, quasi-inverse, and maximum recovery. Most notably, we show that for the class of mappings considered in [Fagin 2007; 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 the above example, a maximum recovery for mapping \mathcal{M}_{E-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 paper. We start by introducing the terminology used in the paper 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 paper, 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, that 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 $\{R_1, \dots, R_k\}$ 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* $\text{dom}(I)$ of instance I is the set of all elements that occur in any of the relations R_i^I . $\text{Inst}(\mathbf{R})$ is defined to be the set of all instances of \mathbf{R} . Given instances $I, J \in \text{Inst}(\mathbf{R})$, 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; Fagin 2007; Fagin et al. 2008]. 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 $\text{Inst}(\mathbf{S})$ 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 $\text{Inst}(\mathbf{T})$ is defined as usual (instances of \mathbf{T} are constructed by using elements from both \mathbf{C} and \mathbf{N}). In this paper, 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 $\text{Inst}(\mathbf{R}_1) \times \text{Inst}(\mathbf{R}_2)$. 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 $\text{Sol}_{\mathcal{M}}(I)$. The domain of \mathcal{M} , denoted by $\text{dom}(\mathcal{M})$, is defined as the set of instances I such that $\text{Sol}_{\mathcal{M}}(I) \neq \emptyset$. Notice that the symbol $\text{dom}(\cdot)$ is used to denote both the domain of an instance and of a mapping, as this does not create any confusion in the paper. 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, $\mathcal{M}_{12} \circ \mathcal{M}_{23} = \{(I_1, I_3) \mid \exists I_2 : (I_1, I_2) \in \mathcal{M}_{12} \text{ and } (I_2, I_3) \in \mathcal{M}_{23}\}$. 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 paper, 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, $\text{CQ}^=$ is the class of conjunctive queries with equalities and 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, $\text{CQ}^{\neq, \mathbf{C}}$ 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; Fagin et al. 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 $\mathbf{R} = \{R_1, \dots, R_m\}$. Instance I can be represented as an $(\mathbf{R} \cup \{\mathbf{C}(\cdot)\})$ -structure $\mathfrak{A}_I = \langle A, R_1^A, \dots, R_m^A, \mathbf{C}^A \rangle$, 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 $\mathbf{R}_1 = \{S_1, \dots, S_m\}$ 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 $\mathbf{R}_2 = \{T_1, \dots, T_n\}$ 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 paper, 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 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 that, in general, no matter what \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} iff $(I, I) \in \mathcal{M} \circ \mathcal{M}'$ 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 $\mathcal{M}'' \preceq_{\mathcal{M}} \mathcal{M}'$, iff $\mathcal{M} \circ \mathcal{M}' \subseteq \mathcal{M} \circ \mathcal{M}''$.

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 . \square

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} . \square

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} . \square

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} . \square

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} , and that 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}^* be a mapping from \mathbf{R}_2 to \mathbf{R}_1 defined as:

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

Given that \mathcal{M}' is a recovery of \mathcal{M} and $(I_2, J_1) \in \mathcal{M}$, we have that \mathcal{M}^* 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}^* where I_2 appears as the second component, we have that $(I_1, I_2) \notin \mathcal{M} \circ \mathcal{M}^*$. Thus, $\mathcal{M} \circ \mathcal{M}' \not\subseteq \mathcal{M} \circ \mathcal{M}^*$ and, therefore, $\mathcal{M}^* \not\preceq_{\mathcal{M}} \mathcal{M}'$. 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 $\mathcal{M}'' \not\leq_{\mathcal{M}} \mathcal{M}'$, 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. \square

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^* such that $P^{J^*} = \{a\}$. It is easy to see that if a source instance I' contains J^* 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^* is a witness for I under \mathcal{M} , as for every source instance I' , if $J^* \in \text{Sol}_{\mathcal{M}}(I')$, then $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I')$. \square

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 $\text{Sol}_{\mathcal{M}}(I_1) = \text{Sol}_{\mathcal{M}}(I_2)$. 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 $J \in \text{Sol}_{\mathcal{M}}(I)$. 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 $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I)$. We have shown that for every I , if $J \in \text{Sol}_{\mathcal{M}}(I)$ then $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I)$ 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 $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. 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. \square

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 iff 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}^* be a mapping defined as $\{(J_I, I) \mid I \in \text{dom}(\mathcal{M})\}$.

$\text{dom}(\mathcal{M})\}$. It is easy to see that \mathcal{M}^* is a reduced recovery of \mathcal{M} . Furthermore, given that for every $(J, I) \in \mathcal{M}^*$, J is a witness for I under \mathcal{M} , we conclude from Lemma 3.11 that \mathcal{M}^* is a maximum recovery of \mathcal{M} . \square

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. 2005a; Arenas et al. 2004; Fagin et al. 2005; 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; 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 $h : \text{dom}(J_1) \rightarrow \text{dom}(J_2)$ such that, for every $R \in \mathbf{R}$ and every tuple $(a_1, \dots, a_k) \in R^{J_1}$, it holds $(h(a_1), \dots, h(a_k)) \in R^{J_2}$. 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 a st-mapping specified by a 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 $\Sigma = \{P(x, y) \rightarrow \exists z(R(x, z) \wedge R(z, y))\}$. 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 . \square

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} . \square

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 Σ_{st} , Γ_{s} , Γ_{t} be sets of source-to-target, source, and target FO-dependencies, respectively. We say that an st-mapping \mathcal{M} is specified by Σ_{st} , Γ_{s} , and Γ_{t} , and we write $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{\text{st}}, \Gamma_{\text{s}}, \Gamma_{\text{t}})$, if \mathcal{M} is specified by $\Sigma_{\text{st}} \cup \Gamma_{\text{s}} \cup \Gamma_{\text{t}}$. Given that both Γ_{s} and Γ_{t} are sets of domain-independent sentences, we have that $(I, J) \models \Sigma_{\text{st}} \cup \Gamma_{\text{s}} \cup \Gamma_{\text{t}}$ if and only if $(I, J) \models \Sigma_{\text{st}}$, $I \models \Gamma_{\text{s}}$ and $J \models \Gamma_{\text{t}}$. 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^* a maximum recovery of \mathcal{M}_1 . If Γ_s is a set of source FO-dependencies and $\mathcal{M}_2 = \{(I, J) \in \mathcal{M}_1 \mid I \models \Gamma_s\}$, then $\mathcal{M}_2^* = \{(J, I) \in \mathcal{M}_1^* \mid I \models \Gamma_s\}$ is a maximum recovery of \mathcal{M}_2 .*

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

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

Notice that \mathcal{M}_1 in the above lemma 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 $\mathcal{M}_2 = (\mathbf{S}, \mathbf{T}, \Sigma_{\text{st}}, \Gamma_s)$ be an st-mapping, where $\mathbf{S} = \{A(\cdot, \cdot, \cdot)\}$, $\mathbf{T} = \{B(\cdot, \cdot), C(\cdot, \cdot)\}$ and

$$\begin{aligned} \Sigma_{\text{st}} &= \{A(x, y, z) \rightarrow B(x, y) \wedge C(y, z)\}, \\ \Gamma_s &= \{A(x, y, z) \wedge A(x', y, z') \rightarrow z = z'\}. \end{aligned}$$

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

$$B(x, y) \wedge C(y, z) \rightarrow A(x, y, z), \quad (2)$$

$$A(x, y, z) \wedge A(x', y, z') \rightarrow z = z'. \quad (3)$$

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

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 shown to be indispensable for the tractability of some important data exchange problems [Fagin et al. 2005a; Fagin et al. 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 $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{\text{st}}, \Gamma_{\text{s}}, \Gamma_{\text{t}})$ be an st-mapping, where Σ_{st} is a set of FO-TO-CQ st-dependencies, Γ_{s} is a set of source FO-dependencies and Γ_{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 $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{\text{st}}, \Gamma_{\text{s}}, \Gamma_{\text{t}})$ be an st-mapping, where Σ_{st} is a set of FO-TO-CQ dependencies, Γ_{s} is a set of source dependencies and Γ_{t} is the union of a set of egds and a weakly acyclic set of tgds, and let $\mathcal{M}_1 = (\mathbf{S}, \mathbf{T}, \Sigma_{\text{st}}, \Gamma_{\text{t}})$. 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 Σ_{st} is a set of FO-TO-CQ dependencies and Γ_{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 Γ_{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

$J \models \Gamma_t$. Thus, we conclude that $J \in \text{Sol}_{\mathcal{M}_1}(I_2)$ since $(I_2, J_1) \in \mathcal{M}_1$, $J \models \Gamma_t$ 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. \square

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 $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ a 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 $J \in \text{Sol}_{\mathcal{M}}(I_1)$ 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 $J \in \text{Sol}_{\mathcal{M}}(I_2)$, but it is not the case that $\text{Sol}_{\mathcal{M}}(I_1) \subseteq \text{Sol}_{\mathcal{M}}(I_2)$. 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 $J \in \text{Sol}_{\mathcal{M}}(I_3)$, but it is not the case that $\text{Sol}_{\mathcal{M}}(I_1) \subseteq \text{Sol}_{\mathcal{M}}(I_3)$.

(2) CQ-TO-UCQ: Let $\mathbf{S} = \{F(\cdot), G(\cdot), H(\cdot)\}$, $\mathbf{T} = \{R(\cdot), S(\cdot), T(\cdot)\}$ and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ a 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 $J \in \text{Sol}_{\mathcal{M}}(I)$ 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 $J \in \text{Sol}_{\mathcal{M}}(I_1)$ and, therefore, $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ 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 $J_1 \in \text{Sol}_{\mathcal{M}}(I)$ but $J_1 \notin \text{Sol}_{\mathcal{M}}(I_1)$, which contradicts the fact that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$.

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⁻: Let $\mathbf{S} = \{F(\cdot), G(\cdot), H(\cdot)\}$, $\mathbf{T} = \{R(\cdot), S(\cdot)\}$, and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ a st-mapping specified by the following set Σ of CQ-TO-CQ⁻ 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 $J \in \text{Sol}_{\mathcal{M}}(I_1)$ 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 $J \in \text{Sol}_{\mathcal{M}}(I_2)$, but it is not the case that $\text{Sol}_{\mathcal{M}}(I_1) \subseteq \text{Sol}_{\mathcal{M}}(I_2)$. 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 $J \in \text{Sol}_{\mathcal{M}}(I_3)$, but it is not the case that $\text{Sol}_{\mathcal{M}}(I_1) \subseteq \text{Sol}_{\mathcal{M}}(I_3)$.

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. \square

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 $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{\text{st}}, \Gamma_{\text{t}})$ an st-mapping specified by the following set Σ_{st} of st-tgds:

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

and the following set Γ_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 $J \in \text{Sol}_{\mathcal{M}}(I)$ 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 $J \in \text{Sol}_{\mathcal{M}}(I_1)$ and, therefore, $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ 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 $J_1 \in \text{Sol}_{\mathcal{M}}(I)$ but $J_1 \notin \text{Sol}_{\mathcal{M}}(I_1)$, which contradicts the fact that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$.

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. \square

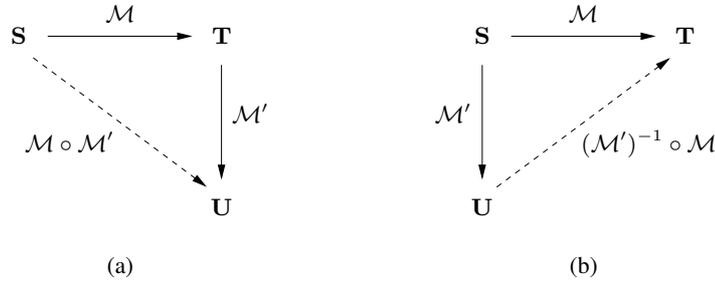


Fig. 1. The schema evolution problem.

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 extent $(\mathcal{M}')^{-1} \circ \mathcal{M}$ 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}, \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}^* 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}^* 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}^* 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}^* , that is, $\text{Sol}_{\mathcal{M}}(I) = \text{Sol}_{\mathcal{M}^*}(J)$. Or, at least, one would expect that none of the instances in $\text{Sol}_{\mathcal{M}}(I)$ is ruled out by \mathcal{M}^* when mapping data from J , that is, $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}^*}(J)$. In this section, we use this simple criterion to compare different solutions for the schema evolution problem.

To formalize the criterion described above, 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 $\bar{x} = (x_1, \dots, x_k)$ and $\bar{y} = (y_1, \dots, y_\ell)$ are tuples of distinct variables, and for every k -tuple \bar{a} from $\text{dom}(I)$ 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 $I \models S(a, b)$, the above 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 $\text{dom}(I)$ for which I satisfies formula $S(x, y)$, we conclude that $T^J = \{(a, n_1), (c, n_3)\}$ and $U^J = \{(b, n_2, n_1), (d, n_4, n_3)\}$. \square

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 mentioned above is formalized as follows: A mapping \mathcal{M}^* 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:

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

The previous criterion also suggests a way to compare alternative solutions for the schema evolution problem; the closer the space of solutions $\text{Sol}_{\mathcal{M}^*}(\text{chase}_{\Sigma'}(I))$ is to $\text{Sol}_{\mathcal{M}}(I)$ the better is \mathcal{M}^* 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}^* for the schema evolution problem for \mathcal{M} and \mathcal{M}' , and for every instance I of \mathbf{S} , it holds that:

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

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

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}^* is a solution for the schema evolution problem for \mathcal{M} and \mathcal{M}' , we have that $\text{Sol}_{\mathcal{M}}(I') \subseteq \text{Sol}_{\mathcal{M}^*}(\text{chase}_{\Sigma'}(I'))$ and, hence, $J \in \text{Sol}_{\mathcal{M}^*}(\text{chase}_{\Sigma'}(I'))$. 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 $J \in \text{Sol}_{\mathcal{M}^*}(\text{chase}_{\Sigma'}(I))$. This concludes the proof of the proposition. \square

Notice that an *ideal* solution for the schema evolution problem for mappings \mathcal{M} and \mathcal{M}' is a mapping \mathcal{M}^* such that $\text{Sol}_{\mathcal{M}}(I) = \text{Sol}_{\mathcal{M}^*}(\text{chase}_{\Sigma'}(I))$, 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 $(I', J) \in \mathcal{M}$. In [Fagin 2007], Fagin 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 $\overline{\text{Id}}$ as $\{(I_1, I_2) \mid (I_1, I_2) \in \text{Inst}(\mathbf{S}) \times \text{Inst}(\mathbf{S}) \text{ and } I_1 \subseteq I_2\}$, 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 $\mathcal{M} \circ \mathcal{M}' = \overline{\text{Id}}$.

Since it is rare that a schema mapping possesses an inverse [Fagin et al. 2008], Fagin et al. 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* w.r.t. \mathcal{M} , denoted by $I_1 \sim_{\mathcal{M}} I_2$, if $\text{Sol}_{\mathcal{M}}(I_1) = \text{Sol}_{\mathcal{M}}(I_2)$. 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 $\{(I_1, I_2) \in \text{Inst}(\mathbf{S}) \times \text{Inst}(\mathbf{S}) \mid \exists (I'_1, I'_2) : I_1 \sim_{\mathcal{M}} I'_1, I_2 \sim_{\mathcal{M}} I'_2 \text{ and } (I'_1, I'_2) \in \mathcal{M}_1\}$. Then a ts-mapping \mathcal{M}' is a *quasi-inverse* of an st-mapping \mathcal{M} if $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] = \overline{\text{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$.

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 $\mathcal{M} \circ \mathcal{M}' = \text{Id} = \{(I, I) \mid I \in \text{Inst}(\mathbf{S})\}$. 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 $\text{dom}(\mathcal{M})$ 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 $\text{Sol}_{\mathcal{M}}(I) = \emptyset$. We first show that \mathcal{M} is not invertible. By hypothesis, for every mapping $\mathcal{M}' \subseteq \text{Inst}(\mathbf{T}) \times \text{Inst}(\mathbf{S})$, it holds that there is no instance I' such that $(I, I') \in \mathcal{M} \circ \mathcal{M}'$, and, therefore, $\mathcal{M} \circ \mathcal{M}' \neq \overline{\text{Id}}$. 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 $\text{Sol}_{\mathcal{M}}(I) = \emptyset$, it holds that $\text{Sol}_{(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]}(I) = \emptyset$. 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

$\text{Sol}_{\mathcal{M}}(I') = \text{Sol}_{\mathcal{M}}(I) = \emptyset$. Thus, we conclude that $\text{Sol}_{\mathcal{M} \circ \mathcal{M}'}(I') = \emptyset$, which contradicts the fact that $(I', I'_1) \in \mathcal{M} \circ \mathcal{M}'$. On the contrary, we have that $(I, I) \in \overline{\text{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$ since $I \sim_{\mathcal{M}} I$ and $(I, I) \in \overline{\text{Id}}$. Thus, we have that $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] \neq \overline{\text{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$. Therefore, we conclude that \mathcal{M} is not quasi-invertible. \square

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} iff \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 $\emptyset \subsetneq \text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. But if $I_1 \subseteq I_2$, we immediately conclude that $\emptyset \subsetneq \text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ 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. \square

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} iff \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

$\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$, 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 $I'_1 \subseteq I'_2$. 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 $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. 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 $(I_2, I_2) \in \overline{\text{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$. 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 $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$, 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 $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] = \overline{\text{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$, there exist 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. \square

PROOF OF THEOREM 6.4. Recall that \mathcal{M}' is a quasi-inverse of \mathcal{M} if $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] = \overline{\text{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$, 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 $J \in \text{Sol}_{\mathcal{M}}(I)$. Now assume that there is an instance I' such that $J \in \text{Sol}_{\mathcal{M}}(I')$, we must prove that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I')$. Given that $J \in \text{Sol}_{\mathcal{M}}(I')$ 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 $\text{Sol}_{\mathcal{M}}(K_2) \subseteq \text{Sol}_{\mathcal{M}}(K_1)$, and then from $(I', I) \sim_{\mathcal{M}} (K_1, K_2)$ we conclude that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I')$. 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, $I'_2 \in \text{dom}(\mathcal{M})$. 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 $\text{Sol}_{\mathcal{M}}(I''_2) \subseteq \text{Sol}_{\mathcal{M}}(I''_1)$. Thus, given that $(I_1, I_2) \sim_{\mathcal{M}} (I''_1, I''_2)$, we conclude that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. 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 $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$, 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) above), 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) above). 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 $(I, I) \notin \mathcal{M} \circ \mathcal{M}'$. \square

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 $J_1 \in \text{Sol}_{\mathcal{M}}(I_2)$, and then $J_1 \in \text{Sol}_{\mathcal{M}}(I_1)$. 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 $J \in \text{Sol}_{\mathcal{M}}(I_1)$ 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 . First, assume that $b_2 = b_3$. Then we have 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 $J_1 \in \text{Sol}_{\mathcal{M}}(I_2)$, but it is not the case that $\text{Sol}_{\mathcal{M}}(I_1) \subseteq \text{Sol}_{\mathcal{M}}(I_2)$, 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 $J_1 \in \text{Sol}_{\mathcal{M}}(I_3)$, but it is not the case that $\text{Sol}_{\mathcal{M}}(I_1) \subseteq \text{Sol}_{\mathcal{M}}(I_3)$, 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 $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ then $I_1 \subseteq I_2$. For the sake of contradiction, assume $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ 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 $J' \in \text{Sol}_{\mathcal{M}}(I_1)$, 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 $J \notin \text{Sol}_{\mathcal{M}}(I_1)$, and then $\text{Sol}_{\mathcal{M}}(I_2) \not\subseteq \text{Sol}_{\mathcal{M}}(I_1)$, which contradicts our initial assumption. By using a similar argument, we can show that if $G^{I_1} \not\subseteq G^{I_2}$ then $\text{Sol}_{\mathcal{M}}(I_2) \not\subseteq \text{Sol}_{\mathcal{M}}(I_1)$, and if $H^{I_1} \not\subseteq H^{I_2}$ then $\text{Sol}_{\mathcal{M}}(I_2) \not\subseteq \text{Sol}_{\mathcal{M}}(I_1)$, which also lead to a contradiction.

In the proof of Proposition 4.8, we show that the above 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 the 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 above 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 above discussion and the fact that both mappings satisfy the subset property. This concludes the proof of the proposition. \square

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 iff \mathcal{M} has a maximum recovery and satisfies the subset property.
- (2) \mathcal{M} is quasi-invertible iff \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 below.

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 $J \in \text{Sol}_{\mathcal{M}}(I_1)$ such that, for every instance I_2 such that $J \in \text{Sol}_{\mathcal{M}}(I_2)$, 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 $J \in \text{Sol}_{\mathcal{M}}(I_1)$ such that, for every source instance I_2 such that $J \in \text{Sol}_{\mathcal{M}}(I_2)$, it is the case that $\text{Sol}_{\mathcal{M}}(I_1) \subseteq \text{Sol}_{\mathcal{M}}(I_2)$. 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 $J \in \text{Sol}_{\mathcal{M}}(I)$ that satisfy condition (3), and $\mathcal{M}' = \{(J, I) \mid I \text{ is a source instance and } J \in U_I\}$. 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 $(\mathcal{M} \circ \mathcal{M}')[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}] = \overline{\text{Id}}[\sim_{\mathcal{M}}, \sim_{\mathcal{M}}]$. 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 $J \in \text{Sol}_{\mathcal{M}}(K_1)$, 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. \square

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 $J \in \text{Sol}_{\mathcal{M}}(I_1)$ such that, for every instance I_2 such that $J \in \text{Sol}_{\mathcal{M}}(I_2)$, 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 $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$, 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 $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$, 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 $J \in \text{Sol}_{\mathcal{M}}(I)$ that satisfy condition (3), and let $\mathcal{M}' = \{(J, I) \mid I \text{ is a source instance and}$

$J \in U_I\}$. 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 $J \in \text{Sol}_{\mathcal{M}}(I_1)$, 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. \square

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 iff 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 sub-classes of GLAV mappings, namely GAV mappings, which essentially corresponds 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 $R(x_1, \dots, x_k) \rightarrow \psi(x_1, \dots, x_k)$, where R is a source predicate [Lenzerini 2002]. However, it is possible to reuse a large part of the work in this area as 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. \square

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 above 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 described above 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 above 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 paper 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 above, 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. \square

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 $\bar{x} = (x_1, \dots, x_m)$, $\bar{y} = (y_1, \dots, y_\ell)$ tuples of distinct variables, and for every m -tuple \bar{a} of elements from $\text{dom}(I)$ 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 $J_I = \text{chase}_\Sigma(I)$. In [Fagin et al. 2005a], the authors prove several properties of $\text{chase}_\Sigma(I)$. In particular, the authors showed 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 $Q(\text{chase}_\Sigma(I))$ that only contains constant values. We denote this last set of tuples by $Q(\text{chase}_\Sigma(I))_\downarrow$. Thus, we have that $\text{certain}_{\mathcal{M}}(Q, I) = Q(\text{chase}_\Sigma(I))_\downarrow$. Notice that if Q' is a rewriting over the source of a conjunctive query Q , then it holds that $Q'(I) = Q(\text{chase}_\Sigma(I))_\downarrow$.

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 as conclusions of different dependencies may be related [Fagin 2007]; a 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:

$$\varphi_1(x_1, x_2) \rightarrow \exists v(P(x_1, v) \wedge R(v, x_2)), \quad (4)$$

$$\varphi_2(y_1, y_2) \rightarrow P(y_1, y_2), \quad (5)$$

$$\varphi_3(z_1, z_2) \rightarrow R(z_1, z_2), \quad (6)$$

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:

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

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 the above 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:

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

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

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

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 $\bar{x} = (x_1, \dots, x_k)$, then $\mathbf{C}(\bar{x})$ is a shorthand for $\mathbf{C}(x_1) \wedge \dots \wedge \mathbf{C}(x_k)$.

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 CQ^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 CQ^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^* an instance of \mathbf{T} such that $(I_1, J^*) \in \mathcal{M}$ and $(J^*, I_2) \in \mathcal{M}'$. 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 $J^* \models \exists \bar{y} \psi(\bar{a}, \bar{y})$. 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 $J^* \models \exists \bar{y} \psi(\bar{a}, \bar{y})$, \bar{a} is a tuple of constant values, and $(J^*, I_2) \models \Sigma'$, 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. \square

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 notice 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 CQ^C-TO-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 QUERYREWRITING(\mathcal{M}, Q) in Lemma 7.2 has as output a query Q' in 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 CQC-TO-UCQ^\equiv 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 CQC-TO-UCQ^\equiv ts-dependencies that specifies a maximum recovery of \mathcal{M}_n is of size $\Omega(2^n)$.*

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

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

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 CQC-TO-UCQ^\equiv 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 $i \in \{1, \dots, n\}$. 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^* such that $(I_R, J^*) \models \Sigma_n$ and $(J^*, I_R) \models \Sigma'$. We show first that the domain of J^* is composed only by null values. On the contrary, assume that there exists a constant element $b \in \mathbf{C}$ such that $b \in \text{dom}(J^*)$. Then it holds that $b \in P_k^{J^*}$ for some $k \in \{1, \dots, n\}$. 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 $i \in \{1, \dots, n\}$ with $i \neq k$. The target instance J' where $P_k^{J'} = \{b\}$ and $P_i^{J'} = \emptyset$ for every $i \in \{1, \dots, n\}$ with $i \neq k$, is such that $(I', J') \in \mathcal{M}_n$. Notice that $J' \subseteq J^*$. Now since Σ_n is a set of st-tgds, we know that \mathcal{M}_n is closed-up on the right, obtaining that $(I', J^*) \in \mathcal{M}_n$. Thus, given that $(J^*, I_R) \in \mathcal{M}'$ 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 it must exist a dependency $\sigma \in \Sigma'$ such that J^* satisfies the premise of σ . Assume that this is not the case. Then since Σ' is a set of CQC-TO-UCQ^\equiv formulas, it would be the case that $(J^*, I_\emptyset) \models \Sigma'$, 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^* . 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^* satisfies the premise of σ , there exists an m -tuple \bar{b} such that $J^* \models \varphi(\bar{b})$. We know that $\varphi(\bar{x})$ is a domain independent formula, then it holds that every component of \bar{b} is in $\text{dom}(J^*)$. We have shown before that $\text{dom}(J^*)$ is composed only by nulls and, thus, every component of \bar{b} is a null value. Now, since $(J^*, I_R) \models \sigma$ and $J^* \models \varphi(\bar{b})$, 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 $\Sigma'' \subseteq \Sigma'$ to be the set of all the dependencies σ of the form $\varphi \rightarrow \psi$ such that $J^* \models \varphi$, 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 $\bar{d} = (d_1, \dots, d_n) \in \{0, 1\}^n$, the set of source relation symbols $\mathbf{S}_{\bar{d}} = \{U_1(\cdot), \dots, U_n(\cdot)\}$ 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 $i \in \{1, \dots, n\}$. 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 CQC 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 CQC are monotone and $J_P \subseteq J_{\bar{d}}$, we have that $J_{\bar{d}} \models \theta$ for every CQC 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 $\text{CQ}^=$ 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 $J^* \models \varphi$ for every $\varphi \rightarrow \psi \in \Sigma''$, obtaining that $(J^*, I_\Gamma) \models \Sigma''$. Notice that Σ'' contains all the dependencies of Σ' such that J^* satisfies their premises and, thus, $(J^*, I_\Gamma) \models \Sigma'$. Then since $(I_R, J^*) \models \Sigma_n$ and $(J^*, I_\Gamma) \models \Sigma'$, we have that $(I_R, I_\Gamma) \in \mathcal{M}_n \circ \mathcal{M}'$. We show now that $\text{Sol}_{\mathcal{M}_n}(I_\Gamma) \not\subseteq \text{Sol}_{\mathcal{M}_n}(I_R)$, which contradicts Proposition 3.8. Notice first that for every target instance $J \in \text{Sol}_{\mathcal{M}_n}(I_R)$, there exists an element $c \in \text{dom}(J)$ such that $c \in P_i^J$ for every $i \in \{1, \dots, n\}$. We prove that there exists an instance in $\text{Sol}_{\mathcal{M}_n}(I_\Gamma)$ that does not satisfy this last property. Consider for every $\alpha \in \Gamma$ the target instance $J_\alpha = \text{chase}_{\Sigma_n}(I_\alpha)$, and let $J_\Gamma = \bigcup_{\alpha \in \Gamma} J_\alpha$. It is easy to see that $J_\Gamma \in \text{Sol}_{\mathcal{M}_n}(I_\Gamma)$. Notice that since every $\alpha \in \Gamma$ mentions a proper subset of the relation symbols of $\mathbf{S}_{\bar{d}}$, there exists an index $i \in \{1, \dots, n\}$ such that $A_i^{I_\alpha} = B_i^{I_\alpha} = \emptyset$, and then there exists an index $i \in \{1, \dots, n\}$ such that $P_i^{J_\alpha} = \emptyset$. Moreover, since $\text{dom}(J_\alpha) \cap \text{dom}(J_{\alpha'}) = \emptyset$ for every pair of distinct elements α, α' of Γ , we obtain that there is no element $c \in \text{dom}(J_\Gamma)$ such that $c \in P_i^{J_\Gamma}$ for every $i \in \{1, \dots, n\}$. Thus, $J_\Gamma \notin \text{Sol}_{\mathcal{M}_n}(I_R)$ implying that $\text{Sol}_{\mathcal{M}_n}(I_\Gamma) \not\subseteq \text{Sol}_{\mathcal{M}_n}(I_R)$

which leads to the contradiction mentioned above. 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 . \square

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 in the fact that given a query composed by 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. \square

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 $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$. \square

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. \square

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 $^\neq$ 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 $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{\text{st}}, \Gamma_{\text{s}})$, where Σ_{st} is a set of FO-TO-CQ st-dependencies (from \mathbf{S} to \mathbf{T}) and Γ_{s} is a set of source FO-dependencies (over \mathbf{S}), algorithm `MAXIMUMRECOVERY` can be used to produce a maximum recovery $\mathcal{M}_1^* = (\mathbf{T}, \mathbf{S}, \Sigma_{\text{ts}})$ for st-mapping $\mathcal{M}_1 = (\mathbf{S}, \mathbf{T}, \Sigma_{\text{st}})$, where Σ_{ts} is a set of CQ^{C} -TO-FO ts-dependencies from \mathbf{T} to \mathbf{S} , and then $\mathcal{M}^* = (\mathbf{T}, \mathbf{S}, \Sigma_{\text{ts}}, \Gamma_{\text{s}})$ 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^{C} -TO-FO dependencies. In this section, we study some properties of the language needed to express maximum recoveries, which provide 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. \square

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. \square

From the proof of Proposition 8.2, we obtain that there exists an st-mapping specified by a set of 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-UCQ^{=' , ']} dependencies suffice to specify maximum recoveries for st-mappings specified by FO-TO-CQ dependencies? Theorem 8.3 below 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 paper 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 CQ^{C} -TO- \mathcal{L}' ts-dependencies.

PROOF. The proof of the theorem is given in electronic appendix B.3. \square

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. \square

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 dependencies 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 the above 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} iff \mathcal{M}' is a recovery of \mathcal{M} .*

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

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 **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 **N-connected** if the graph G_I is connected. An instance I_1 is an **N-connected sub-instance** of I , if I_1 is a sub-instance of I and I_1 is **N-connected**. Finally, I_1 is an **N-connected component** of I , if I_1 is an **N-connected sub-instance** of I and there is no **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 **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. \square

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 $\text{chase}_{\Sigma'}(\text{chase}_\Sigma(I))$ which 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 which 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 **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:

$$\varphi = \forall u_1 \dots \forall u_\ell \exists v_1 \dots \exists v_m \psi, \quad (9)$$

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 $\mathbf{T} = \{U_1(\cdot, \cdot, \cdot), \dots, U_\ell(\cdot, \cdot, \cdot)\}$. 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 :

$$\begin{aligned} \varphi_0(x, y) = & R_0(x, x, x) \wedge R_0(x, x, y) \wedge R_0(x, y, x) \wedge R_0(y, x, x) \wedge \\ & R_0(x, y, y) \wedge R_0(y, x, y) \wedge R_0(y, y, x). \end{aligned}$$

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 $i \in \{1, \dots, \ell\}$. 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:

$$\begin{aligned} T(x, y) \wedge \varphi_0(x, y) \wedge \varphi_1(x, y) \wedge \varphi_2(x, y) \wedge \varphi_3(x, y) \rightarrow \\ U_1(x, x, y) \wedge U_1(y, x, y) \wedge \dots \wedge U_\ell(x, x, y) \wedge U_\ell(y, x, y). \end{aligned} \quad (10)$$

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

$$U_1(u_1, x, y) \wedge \dots \wedge U_\ell(u_\ell, x, y) \rightarrow \exists v_1 \dots \exists v_m \theta(u_1, \dots, u_\ell, v_1, \dots, v_m), \quad (11)$$

where $\theta(u_1, \dots, u_\ell, v_1, \dots, v_m)$ is defined as follows. If 3-CNF formula ψ in (9) is equal to $C_1 \wedge \dots \wedge C_k$, where each C_i is a clause, then $\theta = \theta_1 \wedge \dots \wedge \theta_k$, 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

$$\begin{aligned}\Sigma &= \{TV(x, y) \wedge \varphi_0(x, y) \wedge \varphi_1(x, y) \wedge \varphi_2(x, y) \wedge \varphi_3(x, y) \rightarrow \\ &\quad U_1(x, x, y) \wedge U_1(y, x, y) \wedge U_2(x, x, y) \wedge U_2(y, x, y)\}, \\ \Sigma' &= \{U_1(u_1, x, y) \wedge U_2(u_2, x, y) \rightarrow \exists v_1 (R_1(u_1, v_1, u_2) \wedge R_0(u_1, u_2, v_1))\}.\end{aligned}$$

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) $\text{chase}_{\Sigma}(I_1) \models U_1(c_1, a, b) \wedge \dots \wedge U_{\ell}(c_{\ell}, a, b)$, where each c_i is either a or b , and (3) K is generated from $\exists v_1 \dots \exists v_m \theta(c_1, \dots, c_{\ell}, v_1, \dots, v_m)$ 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 $i \in \{1, \dots, m\}$. Given that φ is valid, we know that for the truth assignment σ_1 such that $\sigma_1(u_i) = c_i$, for every $i \in \{1, \dots, \ell\}$, 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 $h(n_i) = \sigma_2(v_i)$ ($i \in \{1, \dots, m\}$) 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 $(c_1, \dots, c_{\ell}) \in \{a, b\}^{\ell}$, there exists a tuple $(d_1, \dots, d_m) \in \{a, b\}^m$ such that $I \models \theta(c_1, \dots, c_{\ell}, d_1, \dots, d_m)$. 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. \square

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-complete¹. 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.*

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

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

As a corollary of Theorem 9.4 and the results in Section 6, we obtain the following undecidability results for maximum recoveries, inverses² 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. \square

10. CONCLUDING REMARKS

In this paper, 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-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, e.g. closed world semantics [Libkin 2006], and for other data models, e.g. XML.

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), pages 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), pages 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), pages 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), pages 13–22.
- BERNSTEIN, P. 2003. Applying Model Management to Classical Meta Data Problems. In 1st Biennial Conference on Innovative Data Systems Research (CIDR).

²The 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.

- 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), pages 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), pages 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), pages 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), pages 109–116.
- FAGIN, R. 1982. Horn Clauses and Database Dependencies. *Journal of the ACM* 29(4), pages 952–985.
- FAGIN, R. 2007. Inverting Schema Mappings. *ACM Transactions on Database Systems* 32(4).
- FAGIN, R., KOLAITIS, P. G., MILLER, R. J., AND POPA, L. 2005a. Data Exchange: Semantics and Query Answering. *Theoretical Computer Science* 336(1), pages 89–124.
- FAGIN, R., KOLAITIS, P. G., AND POPA, L. 2005b. Data Exchange: Getting to the Core. *ACM Transactions on Database Systems* 30(1), pages 174–210.
- FAGIN, R., KOLAITIS, P. G., POPA, L., AND TAN, W.-C. 2005. Composing Schema Mappings: Second-Order Dependencies to the Rescue. *ACM Transactions on Database Systems* 30(4), pages 994–1055.
- FAGIN, R., KOLAITIS, P. G., POPA, L., AND TAN, W. C. 2008. Quasi-Inverses of Schema Mappings. *ACM Transactions on Database Systems* 33(2).
- GOTTLÖB, 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), pages 40–49.
- HALEVY, A. 2000. Theory of Answering Queries using Views. *SIGMOD Record* 29(1), pages 40–47.
- HALEVY, A. 2001. Answering Queries using Views: A Survey. *VLDB Journal* 10(4), pages 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 (editors). *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), pages 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), pages 30–39.
- LENZERINI, M. 2002. Data Integration: A Theoretical Perspective. In Proceedings of the 21st ACM Symposium on Principles of Database Systems (PODS), pages 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), pages 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), pages 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), pages 60–69.
- MELNIK, S. 2004. *Generic Model Management: Concepts and Algorithms*. Lecture Notes in Computer Science, volume 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), pages 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 Journal* 10(2-3), pages 182–198.

Received Month Year; revised Month Year; accepted Month Year

A. PROOFS OF SECTION 7

A.1 Proof of Lemma 7.2

To prove the lemma, we provide an algorithm that given an st-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ such that Σ is a set of FO-TO-CQ dependencies, and a conjunctive query Q over schema \mathbf{T} , computes a query Q' that is a rewriting of Q over the source schema \mathbf{S} .

We first introduce the terminology used in the algorithm. The basic notion used in the algorithm is that of *existential replacement*. In an existential replacement of a formula β , we are allowed to existentially quantify some of the positions of the free variables of β . For example, if $\beta(x_1, x_2, x_3) = P(x_1, x_2) \wedge R(x_2, x_3)$, then two existential replacements of $\beta(x_1, x_2, x_3)$ are $\gamma_1(x_2) = \exists u \exists v (P(u, x_2) \wedge R(x_2, v))$ and $\gamma_2(x_1, x_2, x_3) = \exists z (P(x_1, z) \wedge R(x_2, x_3))$. We note that both γ_1 and γ_2 are implied by β . In an existential replacement, we are also allowed to use the same quantifier for different positions. For example, $\gamma_3(x_2) = \exists w (P(w, x_2) \wedge R(x_2, w))$ is also an existential replacement of β . We note that γ_3 is implied by β if x_1 and x_3 have the same value, that is, $\beta(x_1, x_2, x_3) \wedge x_1 = x_3$ implies γ_3 . In an existential replacement, these equalities are also included. Formally, given a formula $\beta(\bar{x})$ where $\bar{x} = (x_1, \dots, x_k)$ is a tuple of distinct variables, an *existential replacement* of $\beta(\bar{x})$ is a pair of formulas $(\exists \bar{z} \gamma(\bar{x}', \bar{z}), \theta(\bar{x}''))$, where: (1) $\exists \bar{z} \gamma(\bar{x}', \bar{z})$ is obtained from $\beta(\bar{x})$ by existentially quantifying some of the positions of the free variables of $\beta(\bar{x})$, and \bar{z} is the tuple of fresh variables used in these quantifications, (2) $\theta(\bar{x}'')$ is a conjunction of equalities such that $x_i = x_j$ is in θ ($1 \leq i, j \leq k$ and $i \neq j$) if we replace a position with variable x_i and a position with variable x_j by the same variable z from \bar{z} , and (3) \bar{x}' and \bar{x}'' are the tuples of free variables of $\exists \bar{z} \gamma(\bar{x}', \bar{z})$ and $\theta(\bar{x}'')$, respectively. Notice that $\exists \bar{z} \gamma(\bar{x}', \bar{z})$ is a logical consequence of $\beta(\bar{x}) \wedge \theta(\bar{x}'')$. For example, the following are existential replacements of the formula $\beta(x_1, x_2, x_3) = \exists y_1 (R(x_1, x_2, y_1) \wedge T(y_1, x_3, x_2))$:

$$\begin{aligned} & (\exists y_1 (R(x_1, x_2, y_1) \wedge T(y_1, x_3, x_2)), \text{true}), \\ & (\exists z_1 \exists z_2 \exists y_1 (R(z_1, x_2, y_1) \wedge T(y_1, x_3, z_2)), \text{true}), \\ & (\exists z_1 \exists z_2 \exists y_1 (R(z_1, z_1, y_1) \wedge T(y_1, z_2, z_2)), x_1 = x_2 \wedge x_3 = x_2). \end{aligned}$$

In the first existential replacement above, we have replaced no position, thus obtaining the initial formula $\beta(x_1, x_2, x_3)$ and sentence true (this is a valid existential replacement). In the second existential replacement, although we have replaced some positions of free variables by existentially quantified variables z_1 and z_2 , we include sentence true since no positions with distinct variables are replaced by the same variable from (z_1, z_2) .

In the algorithm, we use the following terminology for tuples of variables: $\bar{x} \subseteq \bar{y}$ indicates that every variable in \bar{x} is also mentioned in \bar{y} , (\bar{x}, \bar{y}) is a tuple of variables obtained by placing the variables of \bar{x} followed by the variables of \bar{y} , $f : \bar{x} \rightarrow \bar{y}$ is a substitution that replaces every variable of \bar{x} by a variable of \bar{y} (f is not necessarily a one-to-one function), $f(\bar{x})$ is a tuple of variables obtained by replacing every variable x in \bar{x} by $f(x)$, and if $\bar{x} = (x_1, \dots, x_k)$ and $\bar{y} = (y_1, \dots, y_k)$, we use formula $\bar{x} = \bar{y}$ as a shorthand for $x_1 = y_1 \wedge \dots \wedge x_k = y_k$.

Algorithm QUERYREWRITING(\mathcal{M}, Q)

Input: An st-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ where Σ is a set of FO-TO-CQ dependencies, and a conjunctive query Q over \mathbf{T} .

Output: An FO query Q' that is a rewriting of Q over the source schema \mathbf{S} .

- (1) Assume that Q is given by the formula $\exists \bar{y} \psi(\bar{x}, \bar{y})$.
- (2) Create a set \mathcal{C}_ψ of FO queries as follows. Start with $\mathcal{C}_\psi = \emptyset$ and let m be the number of atoms in $\psi(\bar{x}, \bar{y})$. Then for every $p \in \{1, \dots, m\}$ and tuple $((\sigma_1, k_1), \dots, (\sigma_p, k_p)) \in (\Sigma \times \{1, \dots, m\})^p$ such that $k_1 + \dots + k_p = m$, do the following.
 - (a) Let (ξ_1, \dots, ξ_p) be a tuple obtained from $(\sigma_1, \dots, \sigma_p)$ by renaming the variables of the formulas $\sigma_1, \dots, \sigma_p$ in such a way that the sets of variables of the formulas ξ_1, \dots, ξ_p are pairwise disjoint.
 - (b) Assume that ξ_i is equal to $\varphi_i(\bar{u}_i) \rightarrow \exists \bar{v}_i \psi_i(\bar{u}_i, \bar{v}_i)$, where \bar{u}_i and \bar{v}_i are tuples of distinct variables.
 - (c) For every tuple $(\chi_1(\bar{w}_1, \bar{z}_1), \dots, \chi_p(\bar{w}_p, \bar{z}_p))$, where $\chi_i(\bar{w}_i, \bar{z}_i)$ is a conjunction of k_i (not necessarily distinct) atoms from $\psi_i(\bar{u}_i, \bar{v}_i)$, $\bar{w}_i \subseteq \bar{u}_i$, $\bar{z}_i \subseteq \bar{v}_i$, and such that \bar{w}_i and \bar{z}_i are tuples of distinct variables, do the following.
 - i. Let $\exists \bar{z} \chi(\bar{w}, \bar{z})$ be the formula $\exists \bar{z}_1 \dots \exists \bar{z}_p (\chi_1(\bar{w}_1, \bar{z}_1) \wedge \dots \wedge \chi_p(\bar{w}_p, \bar{z}_p))$ with $\bar{w} = (\bar{w}_1, \dots, \bar{w}_p)$ and $\bar{z} = (\bar{z}_1, \dots, \bar{z}_p)$.
 - ii. Then for every existential replacement $(\exists \bar{s} \exists \bar{z} \gamma(\bar{w}', \bar{z}, \bar{s}), \theta(\bar{w}''))$ of $\exists \bar{z} \chi(\bar{w}, \bar{z})$ (up to renaming of variables in \bar{s}), and for every pair of variable substitutions $f : \bar{x} \rightarrow \bar{x}$ and $g : \bar{w}' \rightarrow \bar{x}$, check whether there exists a variable substitution $h : \bar{y} \rightarrow (\bar{z}, \bar{s})$ such that $\psi(f(\bar{x}), h(\bar{y}))$ and $\gamma(g(\bar{w}'), \bar{z}, \bar{s})$ are syntactically equal (up to reordering of atoms). If this is the case, then add to \mathcal{C}_ψ the following formula:

$$\exists \bar{u}_1 \dots \exists \bar{u}_p \left(\bigwedge_{i=1}^p \varphi_i(\bar{u}_i) \wedge \theta(\bar{w}'') \wedge \bar{x} = f(\bar{x}) \wedge \bar{w}' = g(\bar{w}') \right). \quad (13)$$

- (3) If \mathcal{C}_ψ is nonempty, then let $\alpha(\bar{x})$ be the FO formula constructed as the disjunction of all the formulas in \mathcal{C}_ψ . Otherwise, let $\alpha(\bar{x})$ be false, that is, an arbitrary unsatisfiable formula (with \bar{x} as its tuple of free variables).
- (4) Return the query Q' given by $\alpha(\bar{x})$.

□

Notice that in the algorithm, tuple \bar{x} is the set of free variables of formula (13) since both \bar{w}' and \bar{w}'' are subsets of $(\bar{u}_1, \dots, \bar{u}_p)$. Also notice that since $\psi(f(\bar{x}), h(\bar{y}))$ and $\gamma(g(\bar{w}'), \bar{z}, \bar{s})$ are identical (up to reordering of atoms), f is a function from \bar{x} to \bar{x} , g is a function from \bar{w}' to \bar{x} , and h is a function from \bar{y} to (\bar{z}, \bar{s}) , we have that every variable x in \bar{x} is equal to some variable u in $(\bar{u}_1, \dots, \bar{u}_p)$ since $\bar{x} = f(\bar{x}) \wedge \bar{w}' = g(\bar{w}')$ is a subformula of (13). This implies that formula (13) is domain independent since each formula $\varphi_i(\bar{u}_i)$ is assumed to be domain independent. Thus, we also have that $\alpha(\bar{x})$ and Q' are domain independent.

Example A.1. Assume that Σ is given by dependency σ :

$$\varphi(x_1, x_2) \rightarrow R(x_1, x_1, x_2), \quad (14)$$

where $\varphi(x_1, x_2)$ is an FO formula over the source schema, and that $Q(x_1, x_2, x_3)$ is the conjunctive query $\exists y_1 \psi(x_1, x_2, x_3, y_1)$, where $\psi(x_1, x_2, x_3, y_1) = R(x_1, x_2, y_1) \wedge R(y_1, x_3, x_3)$. Given that $\psi(x_1, x_2, x_3, y_1)$ has two atoms, the algorithm considers the tuples $(\sigma_1, 2)$ from $(\Sigma \times \{1, 2\})^1$ and $((\sigma_1, 1), (\sigma_2, 1))$ from $(\Sigma \times \{1, 2\})^2$, where $\sigma_1 = \sigma_2 = \sigma$, to construct a source rewriting of query $Q(x_1, x_2, x_3)$. We show here how tuple $((\sigma_1, 1), (\sigma_2, 1))$ is processed.

First, the algorithm generates a tuple (ξ_1, ξ_2) from (σ_1, σ_2) by renaming the variables of σ_1 and σ_2 (in such a way that the sets of variables of ξ_1 and ξ_2 are disjoint). Assume that ξ_1 is equal to $\varphi(u_1, u_2) \rightarrow R(u_1, u_1, u_2)$ and ξ_2 equal to $\varphi(u_3, u_4) \rightarrow R(u_3, u_3, u_4)$. The algorithm continues by considering all the tuples $(\chi_1(u_1, u_2), \chi_2(u_3, u_4))$ such that $\chi_1(u_1, u_2)$ and $\chi_2(u_3, u_4)$ are nonempty conjunctions of atoms from the consequents of ξ_1 and ξ_2 , respectively. In this case, the algorithm only needs to consider tuple $(R(u_1, u_1, u_2), R(u_3, u_3, u_4))$. The algorithm uses this tuple to construct formula $\chi(u_1, u_2, u_3, u_4) = R(u_1, u_1, u_2) \wedge R(u_3, u_3, u_4)$, and then looks for all the existential replacements of $\chi(u_1, u_2, u_3, u_4)$ that can be made identical to $\exists y_1 \psi(x_1, x_2, x_3, y_1)$ by substituting some variables. For instance, $(\exists s_1 (R(u_1, u_1, s_1) \wedge R(s_1, u_3, u_4)), u_2 = u_3)$ is one of these existential replacements: $R(g(u_1), g(u_1), s_1) \wedge R(s_1, g(u_3), g(u_4))$ is syntactically equal to $\psi(f(x_1), f(x_2), f(x_3), h(y_1))$, where $f(x_1) = f(x_2) = x_1$, $f(x_3) = x_3$, $g(u_1) = x_1$, $g(u_3) = g(u_4) = x_3$ and $h(y_1) = s_1$. The algorithm uses functions f , g and condition $u_2 = u_3$ from the existential replacement to generate the following formula $\beta(x_1, x_2, x_3)$ (omitting trivial equalities like $x_1 = x_1$):

$$\begin{aligned} \exists u_1 \exists u_2 \exists u_3 \exists u_4 (\varphi(u_1, u_2) \wedge \varphi(u_3, u_4) \wedge \\ u_2 = u_3 \wedge x_2 = x_1 \wedge u_1 = x_1 \wedge u_3 = x_3 \wedge u_4 = x_3). \end{aligned}$$

Formula $\beta(x_1, x_2, x_3)$ is added to \mathcal{C}_ψ . It is important to notice that $\beta(x_1, x_2, x_3)$ represents a way to deduce $\exists y_1 \psi(x_1, x_2, x_3, y_1)$ from $\varphi(x_1, x_2)$, that is, $\beta(x_1, x_2, x_3) \rightarrow \exists y_1 \psi(x_1, x_2, x_3, y_1)$ is a logical consequence of formula (14).

In the last step of the algorithm, an FO formula $\alpha(x_1, x_2, x_3)$ is generated by taking the disjunction of all the formulas in \mathcal{C}_ψ . In particular, formula $\beta(x_1, x_2, x_3)$ above is one of these disjuncts. The algorithm returns $\alpha(x_1, x_2, x_3)$, which is a rewriting over the source of conjunctive query $Q(x_1, x_2, x_3)$. \square

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be an st-mapping with Σ a set of FO-TO-CQ dependencies, Q a conjunctive query over \mathbf{T} , and Q' the output of $\text{QUERYREWRITING}(\mathcal{M}, Q)$. It is straightforward to prove that the algorithm runs in exponential time in the size of \mathcal{M} and Q , and that the size of Q' is exponential in the size of \mathcal{M} and Q . We now prove the correctness of the rewriting algorithm. We need to show that for every instance I of \mathbf{S} , it holds that:

$$Q'(I) = \text{certain}_{\mathcal{M}}(Q, I).$$

In this proof, we assume that Q is given by the formula $\exists \bar{y} \psi(\bar{x}, \bar{y})$, and that Q' is given by the formula $\alpha(\bar{x})$ (that could be false).

We first show that $Q'(I) \subseteq \text{certain}_{\mathcal{M}}(Q, I)$. The proof relies in the following claim.

CLAIM A.2. *The formula $\forall \bar{x} (\alpha(\bar{x}) \rightarrow \exists \bar{y} \psi(\bar{x}, \bar{y}))$ is a logical consequence of Σ .*

PROOF. If $\alpha(\bar{x})$ is false, the property trivially holds. Now, assume that $\alpha(\bar{x})$ is the disjunction of the formulas in the set \mathcal{C}_ψ constructed after step 2 of the algorithm. We show that for every $\beta(\bar{x}) \in \mathcal{C}_\psi$ it holds that $\forall \bar{x} (\beta(\bar{x}) \rightarrow \exists \bar{y} \psi(\bar{x}, \bar{y}))$ is a logical consequence of Σ , which implies that $\forall \bar{x} (\alpha(\bar{x}) \rightarrow \exists \bar{y} \psi(\bar{x}, \bar{y}))$ is a logical consequence of Σ . Assume that $\beta(\bar{x})$ is equal to:

$$\exists \bar{u}_1 \cdots \exists \bar{u}_p \left(\bigwedge_{i=1}^p \varphi_i(\bar{u}_i) \wedge \theta(\bar{w}'') \wedge \bar{x} = f(\bar{x}) \wedge \bar{w}' = g(\bar{w}') \right),$$

where for every $i \in \{1, \dots, p\}$, it holds that $\varphi_i(\bar{u}_i) \rightarrow \exists \bar{v}_i \psi_i(\bar{u}_i, \bar{v}_i)$ is a dependency in Σ . In step 2(c)i of the algorithm, formula $\exists \bar{z} \chi(\bar{w}, \bar{z})$ is defined as $\exists \bar{z}_1 \dots \exists \bar{z}_p (\chi_1(\bar{w}_1, \bar{z}_1) \wedge \dots \wedge \chi_p(\bar{w}_p, \bar{z}_p))$, where $\chi_i(\bar{w}_i, \bar{z}_i)$ is a conjunction of atoms from $\psi_i(\bar{u}_i, \bar{v}_i)$, with $\bar{w}_i \subseteq \bar{u}_i$ and $\bar{z}_i \subseteq \bar{v}_i$. Thus, we have that sentence Φ :

$$\forall \bar{x} (\beta(\bar{x}) \rightarrow \exists \bar{w} (\exists \bar{z} \chi(\bar{w}, \bar{z}) \wedge \theta(\bar{w}'') \wedge \bar{x} = f(\bar{x}) \wedge \bar{w}' = g(\bar{w}')))$$

is a logical consequence of Σ . Given that $(\exists \bar{s} \exists \bar{z} \gamma(\bar{w}', \bar{z}, \bar{s}), \theta(\bar{w}''))$ is an existential replacement of $\exists \bar{z} \chi(\bar{w}, \bar{z})$, we know that $\exists \bar{z} \chi(\bar{w}, \bar{z}) \wedge \theta(\bar{w}'')$ implies $\exists \bar{s} \exists \bar{z} \gamma(\bar{w}', \bar{z}, \bar{s})$. Thus, we have that Φ implies:

$$\forall \bar{x} (\beta(\bar{x}) \rightarrow \exists \bar{w}' (\exists \bar{s} \exists \bar{z} \gamma(\bar{w}', \bar{z}, \bar{s}) \wedge \bar{x} = f(\bar{x}) \wedge \bar{w}' = g(\bar{w}'))).$$

Now, we can safely replace \bar{w}' by $g(\bar{w}')$, and drop the conjunction $\bar{w}' = g(\bar{w}')$ and the existential quantification over \bar{w}' . Then we obtain that sentence:

$$\forall \bar{x} (\beta(\bar{x}) \rightarrow \exists \bar{s} \exists \bar{z} \gamma(g(\bar{w}'), \bar{z}, \bar{s}) \wedge \bar{x} = f(\bar{x}))$$

is a logical consequence of Φ . Thus, given that $\gamma(g(\bar{w}'), \bar{z}, \bar{s})$ is syntactically equal to $\psi(f(\bar{x}), h(\bar{y}))$, we know that $\forall \bar{x} (\beta(\bar{x}) \rightarrow \exists \bar{s} \exists \bar{z} \psi(f(\bar{x}), h(\bar{y})) \wedge \bar{x} = f(\bar{x}))$ is also a consequence of Φ . In this last formula, we can replace $f(\bar{x})$ by \bar{x} and drop the conjunction $\bar{x} = f(\bar{x})$, obtaining $\forall \bar{x} (\beta(\bar{x}) \rightarrow \exists \bar{s} \exists \bar{z} \psi(\bar{x}, h(\bar{y})))$. Since h is a function from \bar{y} to (\bar{z}, \bar{s}) , we have that $\exists \bar{s} \exists \bar{z} \psi(\bar{x}, h(\bar{y}))$ logically implies formula $\exists \bar{y} \psi(\bar{x}, \bar{y})$ (because the variables in \bar{y} are all distinct). We have shown that $\forall \bar{x} (\beta(\bar{x}) \rightarrow \exists \bar{y} \psi(\bar{x}, \bar{y}))$ is a logical consequence of Φ and, therefore, it is a logical consequence of Σ . This concludes the proof of the claim. \square

We prove now that $Q'(I) \subseteq \text{certain}_{\mathcal{M}}(Q, I)$ for every instance $I \in \text{Inst}(\mathbf{S})$, by using the above claim. Let I be an arbitrary instance, and assume that \bar{a} is a tuple of constant values such that $\bar{a} \in Q'(I)$. We need to show that for every $J \in \text{Sol}_{\mathcal{M}}(I)$ it holds that $\bar{a} \in Q(J)$. Since $\bar{a} \in Q'(I)$ we know that $I \models \alpha(\bar{a})$. Now let $J \in \text{Sol}_{\mathcal{M}}(I)$. From the Claim A.2 we know that $\forall \bar{x} (\alpha(\bar{x}) \rightarrow \exists \bar{y} \psi(\bar{x}, \bar{y}))$ is a logical consequence of Σ . Then since $(I, J) \models \Sigma$ and $I \models \alpha(\bar{a})$, it holds that $J \models \exists \bar{y} \psi(\bar{a}, \bar{y})$, which implies that $\bar{a} \in Q(J)$. Thus we have that for every $J \in \text{Sol}_{\mathcal{M}}(I)$ it holds that $\bar{a} \in Q(J)$. This was to be shown.

We now prove that $\text{certain}_{\mathcal{M}}(Q, I) \subseteq Q'(I)$ for every instance I . We first recall the notion of *chase* (introduced in the proof of Theorem 7.3). Let I be an instance of \mathbf{S} . Then $\text{chase}_{\Sigma}(I)$ is an instance of \mathbf{T} constructed with the following procedure. For every dependency $\sigma \in \Sigma$ of the form $\varphi(\bar{x}) \rightarrow \exists \bar{y} \nu(\bar{x}, \bar{y})$, with $\bar{x} = (x_1, \dots, x_m)$, $\bar{y} = (y_1, \dots, y_\ell)$ tuples of distinct variables, and for every m -tuple \bar{a} of elements from $\text{dom}(I)$ 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 $\nu(\bar{a}, \bar{n})$ in $\text{chase}_{\Sigma}(I)$. We say that the conjuncts of $\psi(a_1, \dots, a_m, n_1, \dots, n_\ell)$ included in $\text{chase}_{\Sigma}(I)$ are *generated* (or *justified*) by σ .

We also make use of the notion of \mathbf{N} -connected instances introduced in the Section 9 when proving Proposition 9.1. Recall that an instance I of \mathbf{S} is \mathbf{N} -connected if the following holds. Let $G_I = (V_I, E_I)$ be a graph such that V_I is composed by all the tuples $t \in R^I$ for $R \in \mathbf{S}$, and there is an edge in E_I between tuples t_1 and t_2 if there exists a 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 . We extend these definitions for formulas that are conjunctions of atoms. Let $\varphi(\bar{x})$ be a conjunction of atoms, and \bar{a} a tuple of values in $\mathbf{C} \cup \mathbf{N}$. We say that $\varphi(\bar{a})$ is \mathbf{N} -connected, if the instance that contains exactly the atoms of $\varphi(\bar{a})$ is \mathbf{N} -connected. The definition of \mathbf{N} -connected components of a conjunction of atoms $\psi(\bar{a})$, is defined as for the case of instances. Notice that if I is such that $\text{dom}(I) \subseteq \mathbf{C}$, then every atom in an \mathbf{N} -connected sub-instance of $\text{chase}_\Sigma(I)$ is generated by a single dependency in Σ .

We are ready now to prove that $\text{certain}_{\mathcal{M}}(Q, I) \subseteq Q'(I)$ for every instance I in \mathbf{S} . Let I be an arbitrary instance of \mathbf{S} . We use the following property of $\text{chase}_\Sigma(I)$. Since Q is a conjunctive query, from [Fagin et al. 2005a] (Proposition 4.2) we know that $\text{certain}_{\mathcal{M}}(Q, I) = Q(\text{chase}_\Sigma(I))_\downarrow$, where $Q(\text{chase}_\Sigma(I))_\downarrow$ denotes the set of tuples in $Q(\text{chase}_\Sigma(I))$ composed only by constant values. Thus, in order to prove that $\text{certain}_{\mathcal{M}}(Q, I) \subseteq Q'(I)$, it is enough to prove that $Q(\text{chase}_\Sigma(I))_\downarrow \subseteq Q'(I)$. Next we show this last property.

Recall that Q is defined by formula $\exists \bar{y} \psi(\bar{x}, \bar{y})$ and Q' by $\alpha(\bar{x})$. Assume that \bar{x} is the tuple of distinct variables (x_1, \dots, x_r) and let $\bar{a} = (a_1, \dots, a_r)$ be a tuple of constant values such that $\bar{a} \in Q(\text{chase}_\Sigma(I))_\downarrow$. Then we know that $\text{chase}_\Sigma(I) \models \exists \bar{y} \psi(\bar{a}, \bar{y})$. We need to show that $\bar{a} \in Q'(I)$, that is, we need to show that $I \models \alpha(\bar{a})$. In order to prove this last fact, we show that after step 2 of the algorithm, there exists a formula $\beta(\bar{x}) \in \mathcal{C}_\psi$ such that $I \models \beta(\bar{a})$.

Assume that in formula $\psi(\bar{x}, \bar{y})$, \bar{y} is the tuple of distinct variables (y_1, \dots, y_ℓ) . Since $\text{chase}_\Sigma(I) \models \exists \bar{y} \psi(\bar{a}, \bar{y})$, we know that there exists a tuple $\bar{b} = (b_1, \dots, b_\ell)$ composed by constant and null values, such that $\text{chase}_\Sigma(I) \models \psi(\bar{a}, \bar{b})$. Let $\rho_1(\bar{a}_1, \bar{b}_1), \dots, \rho_p(\bar{a}_p, \bar{b}_p)$ be the \mathbf{N} -connected components of $\psi(\bar{a}, \bar{b})$, and assume that $\rho_i(\bar{a}_i, \bar{b}_i)$ is a conjunction of k_i (not necessarily distinct) atoms. Notice that if $\psi(\bar{x}, \bar{y})$ has m atoms, then $k_1 + \dots + k_p = m$. Without loss of generality, we can assume that $\psi(\bar{a}, \bar{b}) = \rho_1(\bar{a}_1, \bar{b}_1) \wedge \dots \wedge \rho_p(\bar{a}_p, \bar{b}_p)$ (otherwise we can always reorder the atoms in $\psi(\bar{a}, \bar{b})$). Since $\text{chase}_\Sigma(I) \models \psi(\bar{a}, \bar{b})$, we know that for every $i \in \{1, \dots, p\}$, the conjuncts of $\rho_i(\bar{a}_i, \bar{b}_i)$ are included in the same \mathbf{N} -connected sub-instance of $\text{chase}_\Sigma(I)$. Furthermore, as we have noted before, for every set of facts J that forms an \mathbf{N} -connected sub-instance of $\text{chase}_\Sigma(I)$, there exists a sentence in Σ that justifies J . Then there exist p (not necessarily distinct) sentences $(\sigma_1, \dots, \sigma_p) \in \Sigma^p$, such that the atoms in $\rho_i(\bar{a}_i, \bar{b}_i)$ are generated by σ_i . Let (ξ_1, \dots, ξ_p) be a tuple of dependencies obtained by renaming the variables of $(\sigma_1, \dots, \sigma_p)$ in such a way that the set of variables of the formulas ξ_1, \dots, ξ_p are pairwise disjoint. Assume that every ξ_i is of the form $\varphi_i(\bar{u}_i) \rightarrow \exists \bar{v}_i \psi_i(\bar{u}_i, \bar{v}_i)$. Since σ_i generates all the atoms in $\rho_i(\bar{a}_i, \bar{b}_i)$, we know that for every $i \in \{1, \dots, p\}$, there exists a formula $\chi_i(\bar{w}_i, \bar{z}_i)$, and tuples \bar{c}_i and \bar{n}_i of values in \mathbf{C} and \mathbf{N} , respectively, such that $\chi_i(\bar{w}_i, \bar{z}_i)$ is a conjunction of k_i (not necessarily distinct) atoms from $\psi_i(\bar{u}_i, \bar{v}_i)$ with $\bar{w}_i \subseteq \bar{u}_i$ and $\bar{z}_i \subseteq \bar{v}_i$, and such that $\chi_i(\bar{c}_i, \bar{n}_i)$ is syntactically equal to $\rho_i(\bar{a}_i, \bar{b}_i)$, up to reordering of atoms. Without loss of generality we can assume that $\chi_i(\bar{c}_i, \bar{n}_i) = \rho_i(\bar{a}_i, \bar{b}_i)$. Let $\chi(\bar{w}, \bar{z}) = \chi_1(\bar{w}_1, \bar{z}_1) \wedge \dots \wedge \chi_p(\bar{w}_p, \bar{z}_p)$, with $\bar{w} = (\bar{w}_1, \dots, \bar{w}_p) = (w_1, \dots, w_d)$ and $\bar{z} = (\bar{z}_1, \dots, \bar{z}_p) = (z_1, \dots, z_e)$ tuples of distinct variables. Then we have that $\chi(\bar{c}, \bar{n}) = \psi(\bar{a}, \bar{b})$, where $\bar{c} = (\bar{c}_1, \dots, \bar{c}_p)$ is a tuple of values in \mathbf{C} , and $\bar{n} = (\bar{n}_1, \dots, \bar{n}_p)$ is a tuple of values in \mathbf{N} . Given that the conjuncts of $\rho_i(\bar{a}_i, \bar{b}_i)$ are facts in $\text{chase}_\Sigma(I)$, and each $\rho_i(\bar{a}_i, \bar{b}_i) = \chi_i(\bar{c}_i, \bar{n}_i)$ is an \mathbf{N} -connected component of $\psi(\bar{a}, \bar{b})$, we have that \bar{n} is a tuple of distinct values in \mathbf{N} (since tuples \bar{n}_i and \bar{n}_j do not share any values, for every $i \neq j$). Through the rest of the proof, we assume that $\bar{c} = (c_1, \dots, c_d)$ and $\bar{n} = (n_1, \dots, n_e)$, that is, for every $i \in \{1, \dots, d\}$, c_i is the value

assigned to variable w_i , and for every $i \in \{1, \dots, e\}$, n_i is the value assigned to z_i .

Focus now in the *positions* of $\psi(\bar{x}, \bar{y})$. For every $i \in \{1, \dots, r\}$, we call x_i -*position* to a position in $\psi(\bar{x}, \bar{y})$ where variable x_i occurs. Similarly, for every $i \in \{1, \dots, \ell\}$, a y_i -*position* is a position in $\psi(\bar{x}, \bar{y})$ where variable y_i occurs. Since $\psi(\bar{a}, \bar{b})$ and $\chi(\bar{c}, \bar{n})$ are syntactically equal, there is a one-to-one correspondence between the positions in $\psi(\bar{x}, \bar{y})$ and the positions in $\chi(\bar{w}, \bar{z})$. Then we can talk about x_i - or y_i -positions in general when referring to positions in $\psi(\bar{x}, \bar{y})$ or in $\chi(\bar{w}, \bar{z})$. We use this correspondence of positions and the fact that $\psi(\bar{a}, \bar{b}) = \chi(\bar{c}, \bar{n})$, to create an existential replacement, and functions f , g , and h , as in step 2(c)ii of the algorithm.

We know that \bar{a} is a tuple of constant values. Then from $\psi(\bar{a}, \bar{b}) = \chi(\bar{c}, \bar{n})$, we obtain that every element of \bar{n} is equal to an element of \bar{b} . Furthermore, this last fact implies that every variable of \bar{z} occurs in a y_i -position of $\chi(\bar{w}, \bar{z})$, otherwise, it could not be the case that $\psi(\bar{a}, \bar{b}) = \chi(\bar{c}, \bar{n})$. Consider now the variables y_i such that a variable of \bar{w} occurs in a y_i -position of $\chi(\bar{w}, \bar{z})$. Construct an existential replacement of $\exists \bar{z} \chi(\bar{w}, \bar{z})$ where, for every such variable y_i , all the y_i -positions are replaced by an existentially quantified variable s_i . Let $(\exists \bar{s} \exists \bar{z} \gamma(\bar{w}', \bar{z}, \bar{s}), \theta(\bar{w}''))$ be such a replacement of $\exists \bar{z} \chi(\bar{w}, \bar{z})$. Notice that in the formula $\gamma(\bar{w}', \bar{z}, \bar{s})$, every variable of \bar{w}' occurs in an x_i -position. We define now function h as follows. Let $h : \bar{y} \rightarrow (\bar{z}, \bar{s})$ be a function such that, $h(y_i) = z_j$ if z_j occurs in a y_i -position, and $h(y_i) = s_i$ otherwise. Notice that h is well defined since if variable z_j occurs in a y_i -position, then z_j occurs in every y_i -position (given that \bar{n} is a tuple of distinct values of \mathbf{N} , \bar{c} is a tuple of values of \mathbf{C} , and $\chi(\bar{c}, \bar{n}) = \psi(\bar{a}, \bar{b})$). We define now functions $f : \bar{x} \rightarrow \bar{x}$ and $g : \bar{w}' \rightarrow \bar{x}$. For that purpose, we construct first a partition of the set of variables of (\bar{x}, \bar{w}') , and then, we let f and g assign to every variable a representative of its equivalent class. Consider then, for every value a in \bar{a} , the set V_a of all the variables x_i of \bar{x} such that x_i is assigned value a (that is, $a_i = a$), plus all the variables w_j of \bar{w}' such that w_j is assigned value a (that is, $c_j = a$). Note that, since $\chi(\bar{c}, \bar{n}) = \psi(\bar{a}, \bar{b})$ and every variable of \bar{w}' occurs in an x_i -position, sets V_a do form a partition of (\bar{x}, \bar{w}') . Choose as a representative of every equivalent class, the variable x_i with minimum index in the equivalent class. Then let f and g be such that, $f(x_i) = x_j$ if x_j is the representative of V_{a_i} , and similarly $g(w_i) = x_j$ if x_j is the representative of V_{c_i} . By the definition of the existential replacement, and the definitions of functions f , g , and h , and since $\psi(\bar{a}, \bar{b}) = \chi(\bar{c}, \bar{n})$, we have that $\psi(f(\bar{x}), h(\bar{y}))$ and $\gamma(g(\bar{w}'), \bar{z}, \bar{s})$ are syntactically equal (they coincide in every x_i - and y_i -position). Then we know that the formula:

$$\beta(\bar{x}) = \exists \bar{u}_1 \dots \exists \bar{u}_p \left(\bigwedge_{i=1}^p \varphi_i(\bar{u}_i) \wedge \theta(\bar{w}'') \wedge \bar{x} = f(\bar{x}) \wedge \bar{w}' = g(\bar{w}') \right),$$

is added to \mathcal{C}_ψ after step 2 of the algorithm. We claim that $I \models \beta(\bar{a})$.

Next we show that $I \models \varphi_1(\bar{c}_1^*) \wedge \dots \wedge \varphi_p(\bar{c}_p^*) \wedge \theta(\bar{c}'') \wedge \bar{a} = f(\bar{a}) \wedge \bar{c}' = g(\bar{c}')$, where \bar{c}_i^* is a tuple of elements in \mathbf{C} that contains \bar{c}_i , \bar{c}' is the tuple obtained by restricting \bar{c} to the variables of \bar{w}' , and \bar{c}'' is the tuple obtained by restricting \bar{c} to the variables of \bar{w}'' . Notice that an equality $w_j = w_k$ appears in the formula $\theta(\bar{w}'')$ if $j \neq k$ and both w_j and w_k occur in a y_i -position. Then since $\psi(\bar{a}, \bar{b}) = \chi(\bar{c}, \bar{n})$, we know that b_i (the value assigned to y_i) is equal to both c_j and c_k and, thus, $c_j = c_k$ holds. We conclude that $\theta(\bar{c}'')$ holds. Consider now equality $\bar{a} = f(\bar{a})$. We know by the definition of f that $f(x_i) = x_j$, if x_j is the representative of V_{a_i} . Thus, we have that $a_i = a_j$, which implies

that $\bar{a} = f(\bar{a})$ holds. Next consider equality $\bar{c}' = g(\bar{c}')$. We know by the definition of g that $g(w_i) = x_j$, if x_j is the representative of V_{c_i} . Thus, we have that $c_i = a_j$, which implies that $\bar{c}' = g(\bar{c}')$ holds. Finally, given that for every $i \in \{1, \dots, p\}$, formula $\psi_i(\bar{a}_i, \bar{b}_i) = \chi_i(\bar{c}_i, \bar{n}_i)$ is justified by dependency $\varphi_i(\bar{u}_i) \rightarrow \exists \bar{v}_i \psi_i(\bar{v}_i, \bar{w}_i)$, there exists a tuple \bar{c}_i^* that contains the elements in \bar{c}_i and such that $I \models \varphi_i(\bar{c}_i^*)$. We have shown that $I \models \varphi_1(\bar{c}_1^*) \wedge \dots \wedge \varphi_p(\bar{c}_p^*) \wedge \theta(\bar{c}'') \wedge \bar{a} = f(\bar{a}) \wedge \bar{c}' = g(\bar{c}')$, and, hence, $I \models \beta(\bar{a})$.

We have shown that if $\text{chase}_\Sigma(I) \models \exists \bar{y} \psi(\bar{a}, \bar{y})$ for a tuple \bar{a} of constants, then there exists a formula $\beta(\bar{x}) \in \mathcal{C}_\psi$ such that $I \models \beta(\bar{a})$. Thus, since $\alpha(\bar{x})$ is the disjunctions of the formulas in \mathcal{C}_ψ , we have that $I \models \alpha(\bar{a})$. Recall that $\exists \bar{y} \psi(\bar{x}, \bar{y})$ defined query Q and $\alpha(\bar{x})$ defines query Q' . Therefore, if a tuple \bar{a} of constants is such that $\bar{a} \in Q(\text{chase}_\Sigma(I))$ then we have that $\bar{a} \in Q'(I)$, which implies that $Q(\text{chase}_\Sigma(I))_\downarrow \subseteq Q'(I)$ and then $\text{certain}_{\mathcal{M}}(Q, I) \subseteq Q'(I)$ which is the property that we wanted to obtain. This completes the proof of correctness of the algorithm.

A.2 Proof of Lemma 7.6

The following algorithm computes a rewriting of a conjunctive query given by a single atom without existential quantifiers.

Algorithm QUERYREWRITINGATOM(\mathcal{M}, Q)

Input: An st-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ where Σ is a set of FO-TO-CQ dependencies, and a conjunctive query Q given by a single atom over \mathbf{T} without existential quantifiers.

Output: An FO query Q' that is a rewriting of Q over the source schema \mathbf{S} .

- (1) Construct a set Σ' of dependencies as follows. Start with $\Sigma' = \emptyset$. For every dependency $\sigma \in \Sigma$ of the form $\varphi(\bar{u}) \rightarrow \exists \bar{v} \psi(\bar{u}, \bar{v})$ do the following.
 - (a) For every atom $P(\bar{u}')$ that is a conjunct in $\psi(\bar{u}, \bar{v})$ such that $\bar{u}' \subseteq \bar{u}$, add dependency $\varphi'(\bar{u}') \rightarrow P(\bar{u}')$ to Σ' , where $\varphi'(\bar{u}') = \exists \bar{u}'' \varphi(\bar{u})$ with \bar{u}'' the tuple of variables in \bar{u} that are not mentioned in \bar{u}' .
- (2) Rename the variables of the dependencies in Σ' in such a way that the obtained dependencies have pairwise disjoint sets of variables.
- (3) Assume that Q is given by the atom $R(\bar{x})$, where \bar{x} is a tuple of not necessarily distinct variables that are not mentioned in the dependencies of Σ' .
- (4) Create a set \mathcal{C}_R of FO queries as follows. Start with $\mathcal{C}_R = \emptyset$. Then for every dependency $\varphi(\bar{z}) \rightarrow R(\bar{z})$ in Σ' , add formula $\exists \bar{z}(\varphi(\bar{z}) \wedge \bar{z} = \bar{x})$ to \mathcal{C}_R .
- (5) If \mathcal{C}_R is nonempty, then let $\alpha(\bar{x})$ be the FO formula constructed as the disjunction of all the formulas in \mathcal{C}_R . Otherwise, let $\alpha(\bar{x})$ be false, that is, an arbitrary unsatisfiable formula (with \bar{x} as its tuple of free variables).
- (6) Return the query Q' given by $\alpha(\bar{x})$. □

It is straightforward to see that the algorithm runs in time $O(\|\Sigma\|^2)$ in the general case, and in time $O(\|\Sigma\|)$ if Σ is a set of full FO-TO-CQ dependencies, each dependency with a single atom in its conclusion. Just notice that in the latter case, the set Σ' constructed in the step 1 of the algorithm is of size linear in the size of Σ . The proof of correctness follows directly from the correctness of algorithm QUERYREWRITING of Lemma 7.2. Just observe that if the input of the algorithm QUERYREWRITING is a query Q given by the single atom $R(\bar{x})$ with no existentially quantified variables, then in the step 2 of the algorithm

the parameter m is equal to 1. Also notice that an atom with existentially quantified variables cannot be transformed into $R(\bar{x})$ by applying existential replacements and variable substitutions.

B. PROOFS OF SECTION 8

B.1 Proof of Proposition 8.1

Let $\mathbf{S} = \{P(\cdot), R(\cdot)\}$, $\mathbf{T} = \{T(\cdot)\}$ and Σ be the following set of st-tgds:

$$\begin{aligned} P(x) &\rightarrow \exists y T(y), \\ R(x) &\rightarrow T(x). \end{aligned}$$

Assume that \mathcal{M}' is a recovery of \mathcal{M} that is specified by a set of FO-sentences over $\mathbf{S} \cup \mathbf{T}$. Next we show that \mathcal{M}' is not a maximum recovery of \mathcal{M} .

On the contrary, assume that \mathcal{M}' is a maximum recovery of \mathcal{M} . Let I be an instance of \mathbf{S} such that $P^I = \{a\}$ and $R^I = \emptyset$, where a is an arbitrary element of \mathbf{C} . Since \mathcal{M}' is a recovery of \mathcal{M} , there exists an instance J of \mathbf{T} such that $(I, J) \in \mathcal{M}$ and $(J, I) \in \mathcal{M}'$. We consider two cases.

- First, assume that J mentions an element $b \in \mathbf{C}$, that is not necessarily distinct from a . Then we have that $(I', J) \in \mathcal{M}$, where I' is an instance of \mathbf{S} such that $P^{I'} = \emptyset$ and $R^{I'} = \{b\}$. Thus, given that $(J, I) \in \mathcal{M}'$, we have that $(I', I) \in \mathcal{M} \circ \mathcal{M}'$, which implies that $\emptyset \subsetneq \text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I')$ by Proposition 3.8. Let J' be an instance of \mathbf{T} defined as $T^{J'} = \{n\}$, where n is an arbitrary element of \mathbf{N} . We have that $(I, J') \in \mathcal{M}$ and $(I', J') \notin \mathcal{M}$, which contradicts the fact that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I')$.
- Second, assume that J does not mention any element from \mathbf{C} . Assume that $\text{dom}(J) = \{n_1, \dots, n_k\}$, and let f be a function defined as $f(n_i) = b_i$, where each b_i is an element of \mathbf{C} that is distinct from a and $b_i \neq b_j$ for $i \neq j$. Let J^* be the target instance that results from replacing every value n_i by b_i . It is easy to see that $(I, J^*) \in \mathcal{M}$. Let g be a function with domain $\{a, n_1, \dots, n_k\}$ defined as $g(a) = a$ and $g(n_i) = f(n_i)$. We have that g is an isomorphism from (J, I) to (J^*, I) when we consider these instances as structures over $\mathbf{S} \cup \mathbf{T}$ ³. Thus, given that \mathcal{M}' is specified by a set of FO-sentences over $\mathbf{S} \cup \mathbf{T}$, we conclude that $(J^*, I) \in \mathcal{M}'$. Therefore, there exists an instance J^* of \mathbf{T} such that $(I, J^*) \in \mathcal{M}$, $(J^*, I) \in \mathcal{M}'$ and J^* mentions elements of \mathbf{C} . This leads to a contradiction, as we show in the previous case. This concludes the proof of the proposition.

B.2 Proof of Proposition 8.2

Let $\mathbf{S} = \{S(\cdot, \cdot)\}$, $\mathbf{T} = \{T(\cdot)\}$ and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be an st-mapping specified by the following set Σ of CQ[≠]-TO-CQ dependencies:

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

Next we show that \mathcal{M} has no maximum recovery specified by a set of FO^C-TO-UCQ dependencies.

For the sake of contradiction, assume that $\mathcal{M}^* = (\mathbf{T}, \mathbf{S}, \Sigma^*)$ is a maximum recovery of \mathcal{M} , where Σ^* is a set of FO^C-TO-UCQ dependencies. Let I_1 be a source instance

³Notice that if we consider (J, I) and (J^*, I) as structures over $\mathbf{S} \cup \mathbf{T} \cup \{\mathbf{C}(\cdot)\}$, then g is not an isomorphism from (J, I) to (J^*, I) .

such that $S^{I_1} = \{(a, b)\}$, where a, b are arbitrary elements of \mathbf{C} and $a \neq b$. Given that \mathcal{M}^* is a recovery of \mathcal{M} , there exists a target instance J_1 such that $(I_1, J_1) \in \mathcal{M}$ and $(J_1, I_1) \in \mathcal{M}^*$.

Since \mathcal{M}^* is a maximum recovery of \mathcal{M} , there exists at least one dependency $\varphi(\bar{x}) \rightarrow \psi(\bar{x}) \in \Sigma^*$ such that $J_1 \models \varphi(\bar{c})$, where \bar{c} is a nonempty tuple of elements from \mathbf{C} . On the contrary, assume that this is not the case. Then given that $(J_1, I_1) \models \Sigma^*$ and Σ^* is a set of $\text{FO}^{\mathbf{C}}\text{-TO-UCQ}$ dependencies, we conclude that either $(J_1, I_\emptyset) \models \Sigma^*$, where I_\emptyset is the empty source instance, or $(J_1, I_2) \models \Sigma^*$, where I_2 is a source instance such that $S^{I_2} = \{(c, c)\}$ ($c \in \mathbf{C}, c \neq a, c \neq b$). The former case contradicts Proposition 3.8 since $(I_1, I_\emptyset) \in \mathcal{M} \circ \mathcal{M}^*$ and $\text{Sol}_{\mathcal{M}}(I_\emptyset) \not\subseteq \text{Sol}_{\mathcal{M}}(I_1)$, while the latter case contradicts the same proposition since $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}^*$ and $\text{Sol}_{\mathcal{M}}(I_2) \not\subseteq \text{Sol}_{\mathcal{M}}(I_1)$.

Given that there exists at least one dependency $\varphi(\bar{x}) \rightarrow \psi(\bar{x}) \in \Sigma^*$ such that $J_1 \models \varphi(\bar{c})$, where \bar{c} is a nonempty tuple of elements from \mathbf{C} , and given that $S^{I_1} = \{(a, b)\}$, we have that $\text{dom}(J_1) \cap \mathbf{C} = \{a\}$. On the contrary, assume that this is not the case. First, suppose that $a \notin \text{dom}(J_1) \cap \mathbf{C}$. Then given that $J_1 \models \varphi(\bar{c})$ and $\text{dom}(I_1) = \{a, b\}$, we conclude that every element of \bar{c} is equal to b . Thus, since $(J_1, I_1) \models \Sigma^*$, $S^{I_1} = \{(a, b)\}$ and Σ^* is a set of $\text{FO}^{\mathbf{C}}\text{-TO-UCQ}$ dependencies, we conclude that $(J_1, I_3) \models \Sigma^*$, where I_3 is a source instance such that $S^{I_3} = \{(b, b)\}$. Therefore, given that $(I_1, J_1) \in \mathcal{M}$, we have that $(I_1, I_3) \in \mathcal{M} \circ \mathcal{M}^*$, which implies by Proposition 3.8 that $\text{Sol}_{\mathcal{M}}(I_3) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. But if J_\emptyset is the empty target instance, then $J_\emptyset \in \text{Sol}_{\mathcal{M}}(I_3)$ and $J_\emptyset \notin \text{Sol}_{\mathcal{M}}(I_1)$, which leads to a contradiction. Second, suppose that $b \in \text{dom}(J_1) \cap \mathbf{C}$. Given that $J_1 \models \varphi(\bar{c})$, $a \in \text{dom}(J_1) \cap \mathbf{C}$ and $\text{dom}(I_1) = \{a, b\}$, we have that every element of \bar{c} is equal to either a or b . Let \bar{c}' be a tuple generated from \bar{c} by replacing a by b and b by a . Given that a and b are indistinguishable in J_1 , we conclude that $J_1 \models \varphi(\bar{c}')$, which implies that $\{(a, b)\} \subsetneq S^{I_1}$ and, thus, contradicts the definition of I_1 . Third, assume that $b \notin \text{dom}(J_1) \cap \mathbf{C}$. Then there exists $d \in \text{dom}(J_1) \cap \mathbf{C}$ such that $d \neq a$ and $d \neq b$. Thus, given that $J_1 \models \varphi(\bar{c})$, $a \in \text{dom}(J_1) \cap \mathbf{C}$ and $\text{dom}(I_1) = \{a, b\}$, we have that every element of \bar{c} is equal to a . Let \bar{c}'' be a tuple generated from \bar{c} by replacing a by d . Given that a and d are indistinguishable in J_1 , we conclude that $J_1 \models \varphi(\bar{c}'')$, which implies that $\{(a, b)\} \subsetneq S^{I_1}$ and, thus, contradicts the definition of I_1 .

Given that $\text{dom}(J_1) \cap \mathbf{C} = \{a\}$, $(J_1, I_1) \models \Sigma^*$ and Σ^* is a set of $\text{FO}^{\mathbf{C}}\text{-TO-UCQ}$ dependencies, we conclude that $(J_1, I_4) \models \Sigma^*$, where I_4 is a source instance such that $S^{I_4} = \{(a, a)\}$. Thus, given that $(I_1, J_1) \in \mathcal{M}$, we have that $(I_1, I_4) \in \mathcal{M} \circ \mathcal{M}^*$, which implies by Proposition 3.8 that $\text{Sol}_{\mathcal{M}}(I_4) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$. But if J_\emptyset is the empty target instance, then $J_\emptyset \in \text{Sol}_{\mathcal{M}}(I_4)$ and $J_\emptyset \notin \text{Sol}_{\mathcal{M}}(I_1)$, which leads to a contradiction. This concludes the proof of the proposition.

B.3 Proof of Theorem 8.3

Assume that there exists a nontrivial sentence Φ in \mathcal{L} that is not expressible in \mathcal{L}' , and let \mathbf{S} be the schema of Φ . We define an st-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ as follows. We let P be a fresh relation name, $\mathbf{T} = \{P(\cdot)\}$ and

$$\Sigma = \{\Phi \rightarrow \exists x P(x)\}.$$

For the sake of contradiction, assume that there exists a maximum recovery $\mathcal{M}^* = (\mathbf{T}, \mathbf{S}, \Sigma^*)$ of \mathcal{M} , where Σ^* is a nonempty set of $\text{CQ-TO-}\mathcal{L}'$ dependencies from $\{P(\cdot), \mathbf{C}(\cdot)\}$ to \mathbf{S} . Furthermore, assume that Σ^* contains the following dependencies:

$$\begin{array}{lll} \alpha_i & \rightarrow & \beta_i & 1 \leq i \leq \ell, \\ \gamma_j(x_{j,1}, \dots, x_{j,n_j}) & \rightarrow & \delta_j(x_{j,1}, \dots, x_{j,n_j}) & 1 \leq j \leq m \text{ and } 1 \leq n_j \end{array}$$

To prove the theorem, we consider two cases. In both cases, we denote by J_\emptyset the empty instance for target schema \mathbf{T} .

- (1) Assume that there exist instances I_1 of \mathbf{S} and J_1 of \mathbf{T} such that $I_1 \models \Phi$, $(I_1, J_1) \in \mathcal{M}$, $(J_1, I_1) \in \mathcal{M}^*$ and $\text{dom}(J_1) \subseteq \mathbf{N}$.

Let S be the set of indexes $\{i \mid 1 \leq i \leq \ell \text{ and } J_1 \models \alpha_i\}$. We note that $S \neq \emptyset$. On the contrary, assume that $S = \emptyset$, and let I_2 be an instance of \mathbf{S} such that $I_2 \not\models \Phi$ (such an instance exists since Φ is a nontrivial sentence). Given that $(J_1, I_1) \models \Sigma^*$ and $\text{dom}(J_1) \subseteq \mathbf{N}$, we have $J_1 \not\models \exists x_{i,1} \dots \exists x_{i,n_i} \gamma_i(x_{i,1}, \dots, x_{i,n_i})$ for every $i \in \{1, \dots, m\}$. Thus, given that $S = \emptyset$, we conclude that $(J_1, I_2) \in \mathcal{M}^*$ (in fact, $(J_1, I) \in \mathcal{M}^*$ for every instance I of \mathbf{S}). Therefore, since $(I_1, J_1) \in \mathcal{M}$, we have that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}^*$. Thus, we have by Lemma 3.11 that $\text{Sol}_{\mathcal{M}}(I_2) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ since \mathcal{M}^* is a maximum recovery of \mathcal{M} . But $J_\emptyset \in \text{Sol}_{\mathcal{M}}(I_2)$ since $I_2 \not\models \Phi$, and $J_\emptyset \notin \text{Sol}_{\mathcal{M}}(I_1)$ since $I_1 \models \Phi$, which leads to a contradiction.

Let Ψ be the following sentence:

$$\bigwedge_{i \in S} \beta_i.$$

We note that this sentence is well defined since $S \neq \emptyset$. Next we show that Φ is equivalent to Ψ . First, we assume that I is an instance of \mathbf{S} such that $I \models \Phi$, and we prove that $I \models \Psi$. Given that \mathcal{M}^* is a recovery of \mathcal{M} , there exists an instance J of \mathbf{T} such that $(I, J) \in \mathcal{M}$ and $(J, I) \in \mathcal{M}^*$. Given that $P^J \neq \emptyset$ and $\text{dom}(J_1) \subseteq \mathbf{N}$, we know that there exists a homomorphism from J_1 to J . Thus, for every $i \in S$, we have that $J \models \alpha_i$ since α_i is a conjunctive query. We conclude that for every $i \in S$, it is the case that $I \models \beta_i$ (since $(J, I) \in \mathcal{M}^*$). Therefore, we have that $I \models \Psi$. Second, we assume that I is an instance of \mathbf{S} such that $I \not\models \Phi$, and we prove that $I \not\models \Psi$. On the contrary, assume that $I \models \Psi$, we have that $(J_1, I) \in \mathcal{M}^*$ and, therefore, $(I_1, I) \in \mathcal{M} \circ \mathcal{M}^*$. We conclude by Lemma 3.11 that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ since \mathcal{M}^* is a maximum recovery of \mathcal{M} . But $J_\emptyset \in \text{Sol}_{\mathcal{M}}(I)$ since $I \not\models \Phi$, and $J_\emptyset \notin \text{Sol}_{\mathcal{M}}(I_1)$ since $I_1 \models \Phi$, which leads to a contradiction.

From the previous paragraph, we have that Φ is equivalent to Ψ . But this contradicts the fact that Φ is not expressible in \mathcal{L}' , since each β_i ($1 \leq i \leq \ell$) is an \mathcal{L}' -sentence and \mathcal{L}' is closed under conjunction.

- (2) Assume that for every instance I_1 of \mathbf{S} , if J_1 is an instance of \mathbf{T} such that $I_1 \models \Phi$, $(I_1, J_1) \in \mathcal{M}$ and $(J_1, I_1) \in \mathcal{M}^*$, then $\text{dom}(J_1) \cap \mathbf{C} \neq \emptyset$. In this case, we consider two sub-cases.

- (2.1) Assume that $m = 0$, that is, $\Sigma^* = \{\alpha_i \rightarrow \beta_i \mid 1 \leq i \leq \ell\}$, and let Ψ be the following sentence:

$$\bigwedge_{i=1}^{\ell} \beta_i.$$

Next we show that Φ is equivalent to Ψ . First, we assume that I is an instance of \mathbf{S} such that $I \models \Phi$, and we prove that $I \models \Psi$. Given that \mathcal{M}^* is a recovery of \mathcal{M} , there exists an instance J of \mathbf{T} such that $(I, J) \in \mathcal{M}$ and $(J, I) \in \mathcal{M}^*$. From the

hypothesis, we have that $\text{dom}(J) \cap \mathbf{C} \neq \emptyset$. Thus, given that each α_i ($1 \leq i \leq \ell$) is a conjunctive query over the vocabulary $\{P(\cdot), \mathbf{C}(\cdot)\}$, we conclude that $J \models \alpha_i$ for every $i \in \{1, \dots, \ell\}$. Therefore, for every $i \in \{1, \dots, \ell\}$, it is the case that $I \models \beta_i$ (since $(J, I) \in \mathcal{M}^*$) and, hence, $I \models \Psi$. Second, we assume that I is an instance of \mathbf{S} such that $I \models \Psi$, and we prove that $I \models \Phi$. On the contrary, assume that $I \not\models \Phi$, and let I_1 be an instance of \mathbf{S} such that $I_1 \models \Phi$ (such an instance exists since Φ is a nontrivial sentence). Given that \mathcal{M}^* is a recovery of \mathcal{M} , there exists an instance J_1 of \mathbf{T} such that $(I_1, J_1) \in \mathcal{M}$ and $(J_1, I_1) \in \mathcal{M}^*$. Thus, given that $I \models \Psi$, we have that $(J_1, I) \in \mathcal{M}^*$ and, therefore, $(I_1, I) \in \mathcal{M} \circ \mathcal{M}^*$. We conclude by Lemma 3.11 that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ since \mathcal{M}^* is a maximum recovery of \mathcal{M} . But $J_\emptyset \in \text{Sol}_{\mathcal{M}}(I)$ since $I \not\models \Phi$, and $J_\emptyset \notin \text{Sol}_{\mathcal{M}}(I_1)$ since $I_1 \models \Phi$, which leads to a contradiction.

From the previous paragraph, we have that Φ is equivalent to Ψ . But this contradicts the fact that Φ is not expressible in \mathcal{L}' , since each β_i ($1 \leq i \leq \ell$) is an \mathcal{L}' -sentence and \mathcal{L}' is closed under conjunction.

(2.2) Assume that $m > 0$, and let Ψ be the following sentence:

$$\left(\bigwedge_{i=1}^{\ell} \beta_i \right) \wedge \left(\bigwedge_{j=1}^m \exists x \delta_j(\underbrace{x, \dots, x}_{n_j \text{ times}}) \right).$$

Next we show that Φ is equivalent to Ψ . First, we assume that I is an instance of \mathbf{S} such that $I \models \Phi$, and we prove that $I \models \Psi$. Given that \mathcal{M}^* is a recovery of \mathcal{M} , there exists an instance J of \mathbf{T} such that $(I, J) \in \mathcal{M}$ and $(J, I) \in \mathcal{M}^*$. From the hypothesis, we have that $\text{dom}(J) \cap \mathbf{C} \neq \emptyset$. Thus, given that each α_i ($1 \leq i \leq \ell$) is a conjunctive query over the vocabulary $\{P(\cdot), \mathbf{C}(\cdot)\}$, we conclude that $J \models \alpha_i$ for every $i \in \{1, \dots, \ell\}$. Furthermore, given that each γ_j ($1 \leq j \leq m$) is a conjunctive query over the vocabulary $\{P(\cdot), \mathbf{C}(\cdot)\}$, we conclude that for every $a \in \text{dom}(J) \cap \mathbf{C}$ and $j \in \{1, \dots, m\}$:

$$J \models \gamma_j(\underbrace{a, \dots, a}_{n_j \text{ times}}).$$

Therefore, given that $(J, I) \in \mathcal{M}^*$, we have that for every $i \in \{1, \dots, \ell\}$, it is the case that $I \models \beta_i$, and for every $j \in \{1, \dots, m\}$, it is the case that

$$I \models \exists x \delta_j(\underbrace{x, \dots, x}_{n_j \text{ times}}).$$

Thus, we have that $I \models \Psi$. Second, we assume that I is an instance of \mathbf{S} such that $I \models \Psi$, and we prove that $I \models \Phi$. On the contrary, assume that $I \not\models \Phi$, and let I_1 be an instance of \mathbf{S} such that $I_1 \models \Phi$ (such an instance exists since Φ is a nontrivial sentence). Given that $I \models \Psi$, there exists an element $a \in \text{dom}(I)$ such that for every $j \in \{1, \dots, m\}$:

$$I \models \delta_j(\underbrace{a, \dots, a}_{n_j \text{ times}}).$$

Thus, if J_a is an instance of \mathbf{T} such that $P^{J_a} = \{a\}$, then $(J_a, I) \in \mathcal{M}^*$, and, therefore, $(I_1, I) \in \mathcal{M} \circ \mathcal{M}^*$ since $(I_1, J_a) \in \mathcal{M}$. From Lemma 3.11, we have that

$\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I_1)$ since \mathcal{M}^* is a maximum recovery of \mathcal{M} . But $J_\emptyset \in \text{Sol}_{\mathcal{M}}(I)$ since $I \not\models \Phi$, and $J_\emptyset \notin \text{Sol}_{\mathcal{M}}(I_1)$ since $I_1 \models \Phi$, which leads to a contradiction.

From the previous paragraph, we have that Φ is equivalent to Ψ . But this contradicts the fact that Φ is not expressible in \mathcal{L}' , since each β_i ($1 \leq i \leq \ell$) is an \mathcal{L}' -sentence and \mathcal{L}' is closed under conjunction, existential quantification and free-variable substitution. This concludes the proof of the theorem.

B.4 Proof of Proposition 8.4

To prove the proposition, we need to introduce the notion of *Ehrenfeucht-Fraïssé game*, which characterizes elementary equivalence in FO (see [Libkin 2004]). Let \mathbf{R} be a relational schema. For every pair I_1, I_2 of instances of \mathbf{R} , tuples $\bar{a} = (a_1, \dots, a_m) \in \text{dom}(I_1)^m$ and $\bar{b} = (b_1, \dots, b_m) \in \text{dom}(I_2)^m$ define a *partial isomorphism* from I_1 to I_2 if the following hold:

- For every $i, j \leq m$, $a_i = a_j$ iff $b_i = b_j$.
- For every k -ary relation symbol $R \in \mathbf{R}$ and every sequence $[i_1, \dots, i_k]$ of not necessarily distinct numbers from $\{1, \dots, m\}$, it holds that $(a_{i_1}, \dots, a_{i_k}) \in R^{I_1}$ iff $(b_{i_1}, \dots, b_{i_k}) \in R^{I_2}$.
- For every $i \leq m$, $a_i \in \mathbf{C}$ iff $b_i \in \mathbf{C}$.

The Ehrenfeucht-Fraïssé game is played by two players, called the *spoiler* and the *duplicator*, on two instances I_1, I_2 of \mathbf{R} . In each round i , the spoiler selects either a point $a_i \in \text{dom}(I_1)$, or $b_i \in \text{dom}(I_2)$, and the duplicator responds by selecting $b_i \in \text{dom}(I_2)$, or $a_i \in \text{dom}(I_1)$, respectively. The duplicator wins after m rounds if the tuples (a_1, \dots, a_m) and (b_1, \dots, b_m) define a partial isomorphism from I_1 to I_2 , otherwise the spoiler wins. We use notation $I_1 \equiv_k I_2$ to indicate that the duplicator has a winning strategy in the k -round game on I_1 and I_2 .

The quantifier rank of an FO formula is the maximum depth of quantifier nesting in it. It is well known that if $I_1 \equiv_k I_2$, then I_1 and I_2 agree on all FO sentences over $\mathbf{R} \cup \{\mathbf{C}(\cdot)\}$ of quantifier rank k [Libkin 2004].

We have the necessary to continue with the proof of Proposition 8.4. In the proof of part (1), we use the following terminology. We say that an instance I of a schema \mathbf{R} is the disjoint union of two instances I_1 and I_2 if $\text{dom}(I_1) \cap \text{dom}(I_2) = \emptyset$ and $R^I = R^{I_1} \cup R^{I_2}$ for every $R \in \mathbf{R}$. Furthermore, we say that $I' \subseteq I$ is a connected component of I if: (a) for every $a, b \in \text{dom}(I')$, there exist tuples $t_1 \in R_1^{I'}, \dots, t_m \in R_m^{I'}$ and elements a_1, \dots, a_{m-1} such that a is mentioned in t_1 , b is mentioned in t_m and a_i is mentioned in t_i and t_{i+1} , for every $i \in \{1, \dots, m-1\}$; and (b) for every $I'' \subseteq I$ such that $I' \subseteq I''$ and I'' satisfies condition (a), it holds that $I' = I''$.

Let $\mathbf{S} = \{P(\cdot, \cdot)\}$, $\mathbf{T} = \{R(\cdot, \cdot), S(\cdot, \cdot)\}$,

$$\Sigma = \{P(x, y) \rightarrow \exists z_1 \exists z_2 \exists z_3 (R(x, z_1) \wedge R(y, z_2) \wedge S(z_1, z_3) \wedge S(z_2, z_3))\},$$

$$\Gamma = \{R(x, y) \wedge R(x, z) \rightarrow y = z,$$

$$S(x, y) \rightarrow x = y,$$

$$R(x, y) \wedge R(y, z) \wedge R(y', z) \rightarrow y = y',$$

$$S(x, y) \wedge R(y, z) \wedge R(y', z) \rightarrow y = y',$$

$$S(y, x) \wedge R(y, z) \wedge R(y', z) \rightarrow y = y',$$

$$R(x, y) \wedge R(x', y) \wedge R(y, z) \rightarrow x = x'\},$$

and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma, \Gamma)$. For the sake of contradiction, assume that $\mathcal{M}^* = (\mathbf{T}, \mathbf{S}, \varphi)$ is a maximum recovery of \mathcal{M} that is specified by an FO sentence φ over signature $\{P(\cdot, \cdot), R(\cdot, \cdot), S(\cdot, \cdot), C(\cdot)\}$.

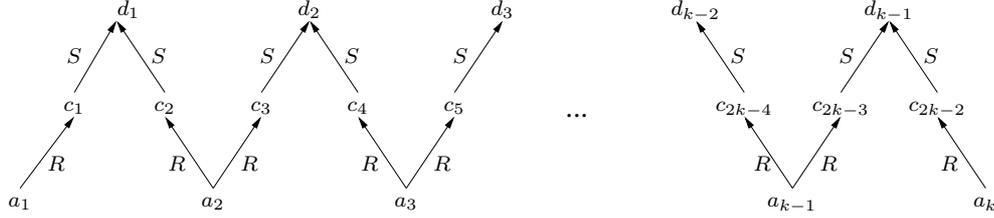
For every $k \geq 2$, let I_k be an instance of \mathbf{S} such that $P^{I_k} = \{(a_1, a_2), \dots, (a_{k-1}, a_k)\}$ and C_k an instance of \mathbf{S} such that $P^{C_k} = \{(a_1, a_2), \dots, (a_{k-1}, a_k), (a_k, a_1)\}$, where $a_i \neq a_j$ ($1 < i < j \leq k$).

CLAIM B.1. *Let J be an instance of \mathbf{T} such that $R^J = \{(a_1, c), \dots, (a_k, c)\}$ and $S^J = \{(c, c)\}$, where $c \neq a_i$ for every $i \in \{1, \dots, k\}$. Then J is a solution for both I_k and C_k .*

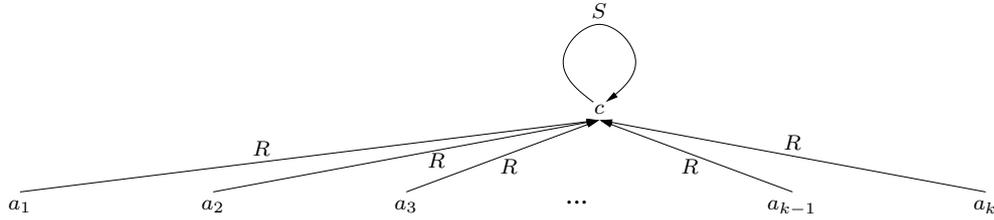
We conclude that every instance I_k is in the domain of \mathcal{M} . Thus, given that \mathcal{M}^* is a maximum recovery of \mathcal{M} , we have by Lemma 3.11 that for every $k \geq 2$, there exists an instance J_k of \mathbf{T} such that $(I_k, J_k) \in \mathcal{M}$, $(J_k, I_k) \in \mathcal{M}^*$ and J_k is a witness for I_k under \mathcal{M} .

CLAIM B.2. *For every $k \geq 2$, if J is a solution for I_k , then J is the disjoint union of two instances J_1 and J_2 , where J_2 could be the empty target instance and (a) $\text{dom}(J_1) = \{a_1, \dots, a_k, b_1, \dots, b_\ell, c\}$, (b) $a_i \neq c$ ($1 \leq i \leq k$), $b_i \neq c$ ($1 \leq i \leq \ell$), $a_i \neq b_j$ ($1 \leq i \leq k$ and $1 \leq j \leq \ell$), $b_i \neq b_j$ ($1 \leq i < j \leq \ell$), (c) $R^{J_1} = \{(a_1, c), \dots, (a_k, c), (b_1, c), \dots, (b_\ell, c)\}$ and $S^{J_1} = \{(c, c)\}$.*

To prove the claim, assume that $(I_k, J) \models \Sigma$. Then we have that J contains at least all the tuples shown in the following figure:



Given that J satisfies dependency $R(x, y) \wedge R(x, z) \rightarrow y = z$, in the above figure $c_2 = c_3$, $c_4 = c_5, \dots, c_{2k-4} = c_{2k-3}$. Thus, given J also satisfies dependency $S(x, y) \rightarrow x = y$, we have that $c_1 = c_2 = \dots = c_{2k-2} = d_1 = \dots = d_{k-1}$. We use c to denote all these elements. We conclude that J contains at least all the tuples shown in the following figure:



Let J_1 be a connected component of J that contains the tuples shown in the figure, and let d be an arbitrary element in this component (d may be equal c or a_j , $1 \leq j \leq k$). Then we have that:

- If $(a_i, d) \in R^{J_1}$, then $d = c$ since J satisfies dependency $R(x, y) \wedge R(x, z) \rightarrow y = z$.
- It could not be the case that $(d, a_i) \in R^{J_1}$ since J satisfies dependency $R(x, y) \wedge R(y, z) \wedge R(y', z) \rightarrow y = y', k \geq 2$ and $a_i \neq a_j$ for every $j \neq i$.
- It could not be the case that $(a_i, d) \in S^{J_1}$ since J satisfies dependency $S(y, x) \wedge R(y, z) \wedge R(y', z) \rightarrow y = y', k \geq 2$ and $a_i \neq a_j$ for every $j \neq i$.
- It could not be the case that $(d, a_i) \in S^{J_1}$ since J satisfies dependency $S(x, y) \wedge R(y, z) \wedge R(y', z) \rightarrow y = y', k \geq 2$ and $a_i \neq a_j$ for every $j \neq i$.
- If $(c, d) \in S^{J_1}$ or $(d, c) \in S^{J_1}$, then $c = d$ since J satisfies dependency $S(x, y) \rightarrow x = y$.
- It could not be the case that $(c, d) \in R^{J_1}$ since J satisfies dependency $R(x, y) \wedge R(x', y) \wedge R(y, z) \rightarrow x = x', k \geq 2$ and $a_i \neq a_j$ for every $j \neq i$.

From these six conditions, we conclude that there exist elements b_1, \dots, b_ℓ such that (a) $\text{dom}(J_1) = \{a_1, \dots, a_k, b_1, \dots, b_\ell, c\}$, (b) $a_i \neq c$ ($1 \leq i \leq k$), $b_i \neq c$ ($1 \leq i \leq \ell$), $a_i \neq b_j$ ($1 \leq i \leq k$ and $1 \leq j \leq \ell$), $b_i \neq b_j$ ($1 \leq i < j \leq \ell$), (c) $R^{J_1} = \{(a_1, c), \dots, (a_k, c), (b_1, c), \dots, (b_\ell, c)\}$ and $S^{J_1} = \{(c, c)\}$. To conclude the proof, we let J_2 be the disjoint union of the remaining connected components of J . Notice that J_2 could be the empty target instance. This concludes the proof of the claim.

Now, for every $k \geq 2$, let I_k^* be an instance of \mathbf{S} such that:

$$P^{I_k^*} = \{(a_1, a_2), \dots, (a_{\lfloor \frac{k}{2} \rfloor - 1}, a_{\lfloor \frac{k}{2} \rfloor})\} \cup \{(a_{\lfloor \frac{k}{2} \rfloor + 1}, a_{\lfloor \frac{k}{2} \rfloor + 2}), \dots, (a_{k-1}, a_k), (a_k, a_{\lfloor \frac{k}{2} \rfloor + 1})\}.$$

By using Claim B.2, it is straightforward to prove the following result.

CLAIM B.3. *There exists a constant $s_0 > 0$ such that, for every $k > s_0$, it holds that $(I_k, J_k) \equiv_k (I_k^*, J_k)$*

Now we have all the necessary ingredients to prove the first part of the proposition. Assume that the quantifier rank of φ is k_0 , and let $s > \max\{k_0, s_0\}$. Next we show that $(I_s, I_s^*) \in \mathcal{M} \circ \mathcal{M}^*$. Given that \mathcal{M}^* is specified by φ and $(J_s, I_s) \in \mathcal{M}^*$, we have that $(J_s, I_s) \models \varphi$. Thus, given that $s > k_0$ and $(J_s, I_s) \equiv_s (J_s, I_s^*)$ (by Claim B.3), we have that $(J_s, I_s^*) \models \varphi$. We conclude that $(J_s, I_s^*) \in \mathcal{M}^*$, which implies that $(I_s, I_s^*) \in \mathcal{M} \circ \mathcal{M}^*$.

Given that \mathcal{M}^* is a maximum recovery of \mathcal{M} and $(I_s, I_s^*) \in \mathcal{M} \circ \mathcal{M}^*$, we know by Proposition 3.8 that $\text{Sol}_{\mathcal{M}}(I_s^*) \subseteq \text{Sol}_{\mathcal{M}}(I_s)$. Let J^* be an instance of \mathbf{T} defined as:

$$\begin{aligned} R^{J^*} &= \{(a_1, c_1), \dots, (a_{\lfloor \frac{s}{2} \rfloor}, c_1)\} \cup \{(a_{\lfloor \frac{s}{2} \rfloor + 1}, c_2), \dots, (a_s, c_2)\}, \\ S^{J^*} &= \{(c_1, c_1), (c_2, c_2)\}, \end{aligned}$$

where $c_1 \neq a_i$ ($1 \leq i \leq s$), $c_2 \neq a_i$ ($1 \leq i \leq s$) and $c_1 \neq c_2$. Then by definition of \mathcal{M} and Claim B.1, we have that $J^* \in \text{Sol}_{\mathcal{M}}(I_s^*)$. Furthermore, we have that $J^* \notin \text{Sol}_{\mathcal{M}}(I_s)$, which contradicts the fact that $\text{Sol}_{\mathcal{M}}(I_s^*) \subseteq \text{Sol}_{\mathcal{M}}(I_s)$. This concludes the proof of the first part of the proposition.

To prove part (2), let $\mathbf{S} = \{P(\cdot, \cdot)\}$, $\mathbf{T} = \{R(\cdot, \cdot)\}$,

$$\Sigma = \{P(x, y) \rightarrow R(x, y)\},$$

$$\Gamma = \{R(x, y) \wedge R(y, z) \rightarrow R(x, z)\},$$

and $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma, \Gamma)$. For the sake of contradiction, assume that $\mathcal{M}^* = (\mathbf{T}, \mathbf{S}, \varphi)$ is a maximum recovery of \mathcal{M} that is specified by an FO sentence φ over signature $\{P(\cdot, \cdot)\}$,

$R(\cdot, \cdot), \mathbf{C}(\cdot)$. Given that \mathcal{M}^* is a maximum recovery of \mathcal{M} , we know from Lemma 3.11 that for every instance I of \mathbf{S} , there exists an instance J_I of \mathbf{T} such that $(I, J_I) \in \mathcal{M}$, $(J_I, I) \in \mathcal{M}^*$ and J_I is a witness for I under \mathcal{M} . Since $(I, J_I) \in \mathcal{M}$, we have that J_I satisfies set Γ of target tgds. Given an instance I of \mathbf{S} , define $\text{TrCl}(I)$ as the transitive closure of I . Then we have that:

CLAIM B.4. *For every source instance I , $R^{J_I} = P^{\text{TrCl}(I)} \cup N_I$, where $P^{\text{TrCl}(I)}$ and N_I are disjoint and for every $(a, b) \in N_I$, it holds that $a \in \mathbf{N}$ or $b \in \mathbf{N}$.*

We now prove the claim. Given that J_I is a solution for I under \mathcal{M} , we have that $P^{\text{TrCl}(I)} \subseteq R^{J_I}$ and, hence, $R^{J_I} = P^{\text{TrCl}(I)} \cup N_I$, where $P^{\text{TrCl}(I)}$ and N_I are disjoint. For the sake of contradiction, assume that there exists a tuple $(a, b) \in N_I$ such that a and b are both elements of \mathbf{C} .

Let I' be an instance of \mathbf{S} defined as $P^{I'} = P^I \cup \{(a, b)\}$. Given that J_I satisfies Γ , we have that $J_I \in \text{Sol}_{\mathcal{M}}(I')$ and, therefore, $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I')$ since J_I is a witness for I under \mathcal{M} . Let J' be an instance of \mathbf{T} defined as $R^{J'} = P^{\text{TrCl}(I)}$. We have that $J' \in \text{Sol}_{\mathcal{M}}(I)$, and given that $P^{\text{TrCl}(I)}$ and N_I are disjoint, we have that $(a, b) \notin P^{\text{TrCl}(I)}$ and, thus, $J' \notin \text{Sol}_{\mathcal{M}}(I')$. This contradicts the fact that $\text{Sol}_{\mathcal{M}}(I) \subseteq \text{Sol}_{\mathcal{M}}(I')$, and concludes the proof of the claim.

To prove the proposition, we need to introduce some terminology. Let k_0 be the quantifier rank of φ . An instance (I, J) of $\mathbf{S} \cup \mathbf{T}$ is \mathcal{M} -cyclic if $\text{dom}(I) = \{a_1, \dots, a_m\}$, $\text{dom}(J) = \{a_1, \dots, a_m\} \cup \{d_1, \dots, d_n\}$, $P^I = \{(a_i, a_{i+1}) \mid 1 \leq i \leq m-1\} \cup \{(a_m, a_1)\}$ and $R^J = \{(a_i, a_j) \mid 1 \leq i, j \leq m\} \cup X$, where X satisfies the following conditions: (1) if $(u, v) \in X$, then $u \in \mathbf{N}$ or $v \in \mathbf{N}$; and (2) if $(a_i, d_j) \in X$, then $(a_k, d_j) \in X$ for every $k \in \{1, \dots, m\}$, and if $(d_j, a_i) \in X$, then $(d_j, a_k) \in X$ for every $k \in \{1, \dots, m\}$. For an \mathcal{M} -cyclic instance (I, J) as above, we say that an instance (I', J') of $\mathbf{S} \cup \mathbf{T}$ is an *amplification* of (I, J) if $\text{dom}(I') = \{b_1, \dots, b_{2m}\}$, $\text{dom}(J') = \{b_1, \dots, b_{2m}\} \cup \{d_1, \dots, d_n\}$, $P^{I'} = \{(b_i, b_{i+1}) \mid 1 \leq i \leq m-1\} \cup \{(b_m, b_1)\} \cup \{(b_{m+i}, b_{m+i+1}) \mid 1 \leq i \leq m-1\} \cup \{(b_{2m}, b_{m+1})\}$ and $R^{J'} = \{(b_i, b_j) \mid 1 \leq i, j \leq 2m\} \cup Y$, where Y satisfies the following conditions: (1) if $(u, v) \in Y$, then $u \in \mathbf{N}$ or $v \in \mathbf{N}$; (2) $X \cap (\{d_1, \dots, d_n\} \times \{d_1, \dots, d_n\}) = Y \cap (\{d_1, \dots, d_n\} \times \{d_1, \dots, d_n\})$; (3) if $(b_i, d_j) \in Y$, then $(b_k, d_j) \in Y$ for every $k \in \{1, \dots, 2m\}$, and if $(d_j, b_i) \in Y$, then $(d_j, b_k) \in Y$ for every $k \in \{1, \dots, 2m\}$.

CLAIM B.5. *There exists a constant $m_0 > 0$ such that, if the domain of an \mathcal{M} -cyclic instance (I, J) contains at least m_0 elements, and (I', J') is an amplification of (I, J) , then $(I, J) \equiv_{k_0} (I', J')$.*

Now we have all the necessary ingredients to prove part (2) of the proposition. Let $m > m_0$ and I_1, I_2 be instances of \mathbf{S} defined as:

$$\begin{aligned} P^{I_1} &= \{(b_i, b_{i+1}) \mid 1 \leq i \leq m-1\} \cup \\ &\quad \{(b_m, b_1)\} \cup \{(b_{m+i}, b_{m+i+1}) \mid 1 \leq i \leq m-1\} \cup \{(b_{2m}, b_{m+1})\}, \\ P^{I_2} &= \{(b_i, b_{i+1}) \mid 1 \leq i \leq m-1\} \cup \{(b_m, b_1)\} \cup \\ &\quad \{(b_{m+i}, b_{m+i+1}) \mid 1 \leq i \leq m-1\} \cup \{(b_{2m}, b_{m+1})\} \cup \{(b_m, b_{m+1})\}. \end{aligned}$$

Next we show that $(I_2, I_1) \in \mathcal{M} \circ \mathcal{M}^*$. Let I be an instance of \mathbf{S} defined as $P^I = \{(a_i, a_{i+1}) \mid 1 \leq i \leq m-1\} \cup \{(a_m, a_1)\}$. Given that $\text{TrCl}(I) = \{(a_i, a_j) \mid 1 \leq i, j \leq$

$m\}$, we have that (I, J_I) is \mathcal{M} -cyclic by Claim B.4 and the fact that J_I satisfies Γ . Thus, by Claim B.5, we have that if (I', J') is an amplification of (I, J_I) , then $(I, J_I) \equiv_{k_0} (I', J')$.

It is straightforward to prove that there is an amplification (I', J') of (I, J_I) such that $I' = I_1$ and $J' \in \text{Sol}_{\mathcal{M}}(I_2)$. Thus, given that \mathcal{M}^* is specified by FO-sentence φ , the quantifier rank of φ is k_0 and $(I, J_I) \equiv_{k_0} (I_1, J')$, we conclude that $(J', I_1) \in \mathcal{M}^*$. Hence, $(I_2, I_1) \in \mathcal{M} \circ \mathcal{M}^*$ since $(I_2, J') \in \mathcal{M}$ and $(J', I_1) \in \mathcal{M}^*$.

Given that \mathcal{M}^* is a maximum recovery of \mathcal{M} and $(I_2, I_1) \in \mathcal{M} \circ \mathcal{M}^*$, we know by Proposition 3.8 that $\text{Sol}_{\mathcal{M}}(I_1) \subseteq \text{Sol}_{\mathcal{M}}(I_2)$. But let J be an instance of \mathbf{T} defined as $R^J = \text{TrCl}(I_1)$. Then we have that $J \in \text{Sol}_{\mathcal{M}}(I_1)$ and $J \notin \text{Sol}_{\mathcal{M}}(I_2)$ since $(b_m, b_{m+1}) \in P^{I_2}$ and $(b_m, b_{m+1}) \notin R^J$. This contradicts the fact that $\text{Sol}_{\mathcal{M}}(I_1) \subseteq \text{Sol}_{\mathcal{M}}(I_2)$, and concludes the proof of the proposition.

C. PROOFS OF SECTION 9

C.1 Proof of Lemma 9.2

(\Rightarrow) Trivial.

(\Leftarrow) Assume that \mathcal{M}' is a recovery of \mathcal{M} . We must show that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$ if and only if $I_1 \subseteq I_2$. By hypothesis, it holds that if $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$ then $I_1 \subseteq I_2$. Now, assume that $I_1 \subseteq I_2$. Since \mathcal{M}' is a recovery of \mathcal{M} , we know that $(I_2, I_2) \in \mathcal{M} \circ \mathcal{M}'$ and then, there exists a target instance J such that $(I_2, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$. Now, given that \mathcal{M} is specified by a set of st-tgds, \mathcal{M} is closed-down on the left and then $(I_1, J) \in \mathcal{M}$. We have that $(I_1, J) \in \mathcal{M}$ and $(J, I_2) \in \mathcal{M}'$, which implies that $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$. This was to be shown.

C.2 Proof of Lemma 9.3

(\Rightarrow) From [Fagin et al. 2005], we know that $\mathcal{M} \circ \mathcal{M}'$ can be specified by a set of st-tgds. Now, from [Fagin 2007] (Proposition 7.2) we know that $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I))$ is a universal solution for I under $\mathcal{M} \circ \mathcal{M}'$, and then $(I, I) \in \mathcal{M} \circ \mathcal{M}'$ if and only if there exists a homomorphism from $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I))$ to I that is the identity on \mathbf{C} . The (\Rightarrow) direction of the proposition follows from the latter condition.

(\Leftarrow) Without loss of generality, assume that each st-tgd in Σ has a single atom in its right-hand side. For the sake of contradiction, suppose that \mathcal{M}' is not a recovery of \mathcal{M} and 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 \mathcal{M}' is not a recovery of \mathcal{M} , there exists an instance I_1 of \mathbf{S} such that $(I_1, I_1) \notin \mathcal{M} \circ \mathcal{M}'$. Let I be an instance of \mathbf{S} . Given that $\text{chase}_{\Sigma}(I)$ is a universal solution for I under \mathcal{M} and $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I))$ is a universal solution for $\text{chase}_{\Sigma}(I)$ under \mathcal{M}' , it is straightforward to prove that if $(I, I') \in \mathcal{M} \circ \mathcal{M}'$, then there exists a homomorphism from $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I))$ to I' that is the identity on \mathbf{C} . Furthermore, if there exists a homomorphism from $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I))$ to an instance I' , then one can conclude that $(\text{chase}_{\Sigma}(I), I') \in \mathcal{M}'$ since $(\text{chase}_{\Sigma}(I), \text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I))) \in \mathcal{M}'$ and $\text{chase}_{\Sigma}(I)$ does not mention any null values as Σ is a set of full st-tgds. Thus, we have that if there exists a homomorphism from $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I))$ to an instance I' , then $(I, I') \in \mathcal{M} \circ \mathcal{M}'$. In particular, from the previous properties, we conclude that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$ if and only if there exists a homomorphism from $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I))$ to I that is the identity on \mathbf{C} . Thus, given that $(I_1, I_1) \notin \mathcal{M} \circ \mathcal{M}'$, there is no homomorphism from $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I_1))$ to I_1 that is the identity on \mathbf{C} , which implies that there exists an \mathbf{N} -connected compo-

nent K_1 of $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I_1))$ such that there is no homomorphism from K_1 to I_1 that is the identity on \mathbf{C} . Given that K_1 is an \mathbf{N} -connected component and Σ is a set of full st-tgds, there exists a dependency $\alpha(\bar{x}) \rightarrow \exists \bar{y} \beta(\bar{x}, \bar{y})$ in Σ' and a tuple \bar{a} of elements from \mathbf{C} such that $\text{chase}_{\Sigma}(I_1) \models \alpha(\bar{a})$ and K_1 is generated from $\exists \bar{y} \beta(\bar{a}, \bar{y})$ when computing $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I_1))$. Assume that $\alpha(\bar{a})$ is equal to $T_1(\bar{a}_1) \wedge \dots \wedge T_n(\bar{a}_n)$. Then for every $i \in \{1, \dots, n\}$, there exists a full st-tgd $\gamma_i(\bar{x}_i) \rightarrow T_i(\bar{x}_i)$ such that $I_1 \models \gamma_i(\bar{a}_i)$. Let I_2 be a sub-instance of I_1 given by the union of all the tuples in the formulas $\gamma_i(\bar{a}_i)$ ($i \in \{1, \dots, n\}$). Then we have that K_1 is an \mathbf{N} -connected component of $\text{chase}_{\Sigma'}(\text{chase}_{\Sigma}(I_2))$ and there is no homomorphism from K_1 to I_2 that is the identity on \mathbf{C} . But by definition of I_2 , we know that $|I_2| \leq k_{\Sigma} \cdot k_{\Sigma'}$, which contradicts our initial assumption.

C.3 Proof of Theorem 9.4

To prove this theorem, we reduce the problem of verifying whether a deterministic Turing Machine (DTM) accepts the empty string to the complement of the problem of verifying whether a schema mapping \mathcal{M}' is a recovery of a schema mapping \mathcal{M} .

Let $M = (Q, \Gamma, q_0, \delta, q_f)$ be a DTM, where Q , Γ , q_0 , δ and q_f are the finite set of states, tape alphabet, initial state, transition function and final state of M . For the sake of simplicity, we assume that $q_0 \neq q_f$, the input alphabet is $\{0, 1\}$ and the tape alphabet Γ is $\{0, 1, \mathbb{B}\}$, where \mathbb{B} is the blank symbol. Furthermore, we assume that the tape of M is infinite to the right and that in each step of M the head has to move either to the right (\mathbb{R}) or to the left (\mathbb{L}), that is, δ is a total function from $(Q \setminus \{q_f\}) \times \Gamma$ to $Q \times \Gamma \times \{\mathbb{L}, \mathbb{R}\}$. Notice that we also assume that no transitions are defined for the final state q_f .

Next we define data exchange settings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$ in such a way that M accepts the empty string if and only if \mathcal{M}' is not a recovery of \mathcal{M} . Relational schemas \mathbf{S} and \mathbf{T} are defined as follows:

$$\begin{aligned} \mathbf{S} &:= \{D(\cdot, \cdot), T(\cdot)\}, \\ \mathbf{T} &:= \{D'(\cdot, \cdot), T'(\cdot), Z(\cdot), O(\cdot), E(\cdot, \cdot, \cdot), L(\cdot, \cdot, \cdot), P(\cdot, \cdot), U(\cdot, \cdot), S(\cdot, \cdot, \cdot), H(\cdot, \cdot, \cdot), \\ &\quad T_0(\cdot, \cdot, \cdot), T_1(\cdot, \cdot, \cdot), T_{\mathbb{B}}(\cdot, \cdot, \cdot)\} \cup \{S_q(\cdot, \cdot) \mid q \in Q\}. \end{aligned}$$

Before defining sets Σ and Σ' of tgds, we give the intended interpretations of the predicates of \mathbf{S} and \mathbf{T} . As it is customary when doing a logical encoding of a Turing Machine, we have a predicate L to store a linear order. Since negation is not allowed in tgds, predicate L is ternary and its third argument is used to indicate whether the first argument is or is not less than the second one. Predicate Z in \mathbf{T} is used to store elements that represent the truth value *false* (or *zero*), while predicate O in \mathbf{T} is used to store elements that represent the truth value *true* (or *one*). Thus, for example, if we want to say that a is less than b according to linear order L , then we add a tuple (a, b, c) to L , where c belongs to O .

As it is also customary when doing a logical encoding of a Turing Machine, we have predicates P and U to store the first and last elements of linear order L , a predicate S to store the successor relation associated with L , a predicate H to store the position of the head of DTM M , predicates S_q ($q \in Q$) to store the state of M and predicates $T_0, T_1, T_{\mathbb{B}}$ to indicate for each cell of the tape of M whether its value is either 0 or 1 or \mathbb{B} , respectively. As for the case of linear order L , all these predicates have an extra argument that is used to indicate whether the predicate is true or false for a particular tuple. Since the equality symbol $=$ is not allowed in st-tgds, we also have an equality predicate E , which is ternary,

as in the previous cases.

To explain the intended interpretations of predicates T of \mathbf{S} and T' of \mathbf{T} , we need to show some of the dependencies in Σ and Σ' . Set Σ contains st-tgds:

$$\begin{aligned} D(x, y) &\rightarrow D'(x, y), \\ T(x) &\rightarrow T'(x) \end{aligned}$$

and Σ' contains ts-tgds:

$$\begin{aligned} D'(x, y) &\rightarrow D(x, y), \\ T'(x) &\rightarrow T(x), \\ Z(x) &\rightarrow T(x), \\ O(x) &\rightarrow T(x). \end{aligned}$$

Thus, if I is an instance of \mathbf{S} and J is a solution for I under $\mathcal{M} \circ \mathcal{M}'$, then $I \subseteq J$. After giving the definitions of Σ and Σ' , we prove that DTM M accepts the empty string if and only if there exists an instance I of \mathbf{S} such that $(I, I) \notin \mathcal{M} \circ \mathcal{M}'$ and, thus, we conclude that M accepts the empty string if and only if \mathcal{M}' is not a recovery of \mathcal{M} .

Let I be an instance of \mathbf{S} and $\text{dom}(D^I)$ the set of all elements mentioned in D^I . Each element in $\text{dom}(D^I)$ is used to denote a position in the tape of M as well as a point of time in the execution of this DTM. Thus, some of the dependencies of Σ should be used to guess a linear order L on $\text{dom}(D^I)$. More precisely, if $\text{dom}_D(x)$ is defined as $\exists y D(x, y) \vee \exists z D(z, x)$, then we should include in Σ an axiom of the form $\text{dom}_D(x) \wedge \text{dom}_D(y) \rightarrow \exists u L(x, y, u)$ stating that for every pair of elements x and y in D , there is a truth value associated with the statement “ x is less than y according to L ”, but with the additional restriction that u belongs to either Z or O . We cannot impose this restriction by using an axiom of the form $\text{dom}_D(x) \wedge \text{dom}_D(y) \rightarrow \exists u (L(x, y, u) \wedge Z(u))$ because we will be explicitly saying that x is not less than y according to L , and the same happens with axiom $\text{dom}_D(x) \wedge \text{dom}_D(y) \rightarrow \exists u (L(x, y, u) \wedge O(u))$. To overcome this problem, we use the fact that T' is a copy of T and all the elements of Z and O belong to T' (since $Z(x) \rightarrow T(x)$ and $O(x) \rightarrow T(x)$ are in Σ'), and we replace the previous axiom by $\text{dom}_D(x) \wedge \text{dom}_D(y) \rightarrow \exists u (L(x, y, u) \wedge T'(u))$, which indeed says that u is a truth value but without explicitly stating whether this value is *true* or *false*.

In the definition of L , we would also like to say that L is defined only for the elements in $\text{dom}(D^I)$, thus avoiding extra elements that are not in D and can behave inadequately. The problem with this is that we cannot include in Σ' an axiom of the form $L(x, y, z) \rightarrow \text{dom}_D(x) \wedge \text{dom}_D(y)$, where $\text{dom}_D(x) = \exists y D(x, y) \vee \exists z D(z, x)$, because this disjunctive sentence is not equivalent to any set of tgds, unlike $\text{dom}_D(x) \wedge \text{dom}_D(y) \rightarrow \exists u (L(x, y, u) \wedge T'(u))$ which is equivalent to a set of four tgds. To overcome this problem, we simply replace disjunction by conjunction in the definitions of $\text{dom}_D(x)$ and $\text{dom}_D(y)$, and thus we replace the previous dependency by:

$$\begin{aligned} \mu(x) \wedge \mu(y) &\rightarrow \exists u (L(x, y, u) \wedge T'(u)), \\ L(x, y, z) &\rightarrow \mu(x) \wedge \mu(y), \end{aligned}$$

where $\mu(x) = \exists y D(x, y) \wedge \exists z D(z, x)$. Hence, predicate L is only defined for the elements that appear in both columns of D .

In order to properly define L , we also need to include the axioms that define L as a linear order. In particular, we need to say that L is connected, that is, for every pair of distinct

elements x and y , we have that x is less than y or y is less than x according to L . Since the equality symbol is not allowed in st-tgds, we use the equality predicate E to express this axiom. Thus, Σ also has to include some dependencies defining E . In particular, we include in Σ an axiom of the form $D(x, y) \rightarrow \exists u (E(x, y, u) \wedge Z(u))$, saying that for every tuple (x, y) in D , we have that x and y are distinct elements. Thus, we also use D to store an inequality relation.

A natural question at this point is what happens if some of the implicit assumptions made above are not satisfied. For example, what happens if D or T is empty, or if D contains a tuple of the form (x, x) , or if Z and O share some values (which implies that some facts could be both true and false), or if T' contains a value which is neither in Z nor in O , or if there is no element a in D satisfying μ ? After defining Σ and Σ' , we make explicit these assumptions, and we show that some of the conditions in Σ and Σ' ensure that, if a source instance I does not satisfy any of these assumptions, then $(I, I) \in \mathcal{M} \circ \mathcal{M}'$. Thus, when checking whether \mathcal{M}' is a recovery of \mathcal{M} , we only need to take into account source instances satisfying these assumptions.

Now we are ready to define sets Σ and Σ' of tgds. For the sake of completeness, we also include some of the dependencies already mentioned.

Copying axioms. Σ contains copying st-tgds:

$$\begin{aligned} D(x, y) &\rightarrow D'(x, y), \\ T(x) &\rightarrow T'(x), \end{aligned}$$

and Σ' contains copying ts-tgds:

$$\begin{aligned} D'(x, y) &\rightarrow D(x, y), \\ T'(x) &\rightarrow T(x). \end{aligned}$$

Definition of predicates Z and O . Let

$$\lambda := \exists x \exists y (D(x, y) \wedge T(x) \wedge T(y)).$$

To define predicates Z and O , we include the following st-tgd in Σ :

$$\lambda \rightarrow \exists u O(u) \wedge \exists v Z(v).$$

This dependency says that if a source instance I satisfies λ , then for every solution J for I , both Z^J and O^J are not empty. Sentence λ is included in this dependency to deal with source instances not satisfying some of the assumptions mentioned above (this will be formalized in Lemmas C.1, C.2 and C.3).

To complete the definition of Z and O , we include the following ts-tgds in Σ' :

$$\begin{aligned} Z(x) &\rightarrow T(x), \\ O(x) &\rightarrow T(x), \\ Z(x) \wedge O(y) &\rightarrow D(x, y). \end{aligned}$$

Definition of equality predicate E . Let

$$\mu(x) := \exists y D(x, y) \wedge \exists z D(z, x).$$

To define the equality predicate E , we include the following st-tgds in Σ :

$$\begin{aligned} \lambda \wedge \mu(x) \wedge \mu(y) &\rightarrow \exists u (E(x, y, u) \wedge T'(u)), \\ \lambda \wedge \mu(x) \wedge \mu(y) \wedge D(x, y) &\rightarrow \exists u (E(x, y, u) \wedge Z(u)). \end{aligned}$$

The first dependency says that for every pair of elements x, y satisfying μ , either x is equal to y or x is not equal to y according to E . The second st-tgd says that for every tuple (x, y) in D , we have that x is different from y (according to E). We also need to state that E is an equivalence relation, which is done by including the following dependencies into Σ' :

$$\begin{aligned} E(x, y, u_1) \wedge Z(u_1) \wedge E(x, y, u_2) \wedge O(u_2) &\rightarrow \exists v D(v, v), \\ E(x, x, u) \wedge Z(u) &\rightarrow \exists v D(v, v), \\ E(x, y, u_1) \wedge O(u_1) \wedge E(y, x, u_2) \wedge Z(u_2) &\rightarrow \exists v D(v, v), \\ E(x, y, u_1) \wedge O(u_1) \wedge E(y, z, u_2) \wedge O(u_2) \wedge E(x, z, u_3) \wedge Z(u_3) &\rightarrow \exists v D(v, v). \end{aligned}$$

The first ts-tgd above says that unless there is a tuple (a, a) in D , it could not be the case that x is equal to y and x is not equal to y according to E . The remaining three dependencies define E as an equivalent relation, provided that D does not contain a tuple (a, a) .

Finally, we also include in Σ' the following ts-tgd:

$$E(x, y, z) \rightarrow \mu(x) \wedge \mu(y),$$

stating that E is only defined for the elements that satisfy μ .

Definition of linear order L . We include the following st-tgd in Σ :

$$\lambda \wedge \mu(x) \wedge \mu(y) \rightarrow \exists u (L(x, y, u) \wedge T'(u)),$$

and we include the following ts-tgds in Σ' :

$$\begin{aligned} L(x, x, u) \wedge O(u) &\rightarrow \exists v D(v, v), \\ L(x, y, u_1) \wedge O(u_1) \wedge L(y, z, u_2) \wedge O(u_2) \wedge L(x, z, u_3) \wedge Z(u_3) &\rightarrow \exists v D(v, v), \\ E(x, y, u_1) \wedge Z(u_1) \wedge L(x, y, u_2) \wedge Z(u_2) \wedge L(y, x, u_3) \wedge Z(u_3) &\rightarrow \exists v D(v, v). \end{aligned}$$

The first dependency above says that L is irreflexive, the second says that L is transitive and the third says that L is connected (unless there is a tuple (a, a) in D). To complete the definition of L , we include in Σ' the following ts-tgd:

$$L(x, y, z) \rightarrow \mu(x) \wedge \mu(y),$$

stating that L is only defined for the elements that satisfy μ , and we also include in Σ' an axiom that states that L is consistent with E (that is, E is a congruence relation for L):

$$\begin{aligned} L(x, y, u_1) \wedge O(u_1) \wedge E(x, x_1, u_2) \wedge O(u_2) \wedge \\ E(y, y_1, u_3) \wedge O(u_3) \wedge L(x_1, y_1, u_4) \wedge Z(u_4) &\rightarrow \exists v D(v, v). \end{aligned}$$

This dependency says that unless there is a tuple (a, a) in D , if x is equal to x_1 and y is equal to y_1 according to E , then x is less than y according to L if and only if x_1 is less than y_1 according to L .

Definition of predicate P. This predicate is used to store the first element of L . To define this predicate, we include in Σ the following st-tgd:

$$\lambda \wedge \mu(x) \rightarrow \exists u_1 \exists u_2 (P(u_1, u_2) \wedge O(u_2)).$$

This st-tgd says that if there exists at least one element satisfying formula μ , and also formula λ is satisfied, then there is a first element for linear order L . Moreover, we also include the following ts-tgds in Σ' :

$$\begin{aligned} P(x, u_1) \wedge O(u_1) \wedge L(y, x, u_2) \wedge O(u_2) &\rightarrow \exists v D(v, v), \\ P(x, u_1) \wedge O(u_1) \wedge E(x, y, u_2) \wedge O(u_2) \wedge P(y, u_3) \wedge Z(u_3) &\rightarrow \exists v D(v, v), \\ P(x, y) &\rightarrow \mu(x). \end{aligned}$$

The first dependency above says that the first element of L does not have a predecessor according to L , while the second says that unless there is a tuple (a, a) in D , if x is equal to y according to E , then x is the first element of L if and only if y is the first element of L (that is, E is a congruence relation for P). Moreover, the last dependency above states that P is only defined for the elements that satisfy μ .

Definition of predicate U. This predicate is used to store the last element of L . To define this predicate, we include the following st-tgd in Σ :

$$\lambda \wedge \mu(x) \rightarrow \exists u_1 \exists u_2 (U(u_1, u_2) \wedge O(u_2)).$$

and we include the following ts-tgds in Σ' :

$$\begin{aligned} U(x, u_1) \wedge O(u_1) \wedge L(x, y, u_2) \wedge O(u_2) &\rightarrow \exists v D(v, v), \\ U(x, u_1) \wedge O(u_1) \wedge E(x, y, u_2) \wedge O(u_2) \wedge U(y, u_3) \wedge Z(u_3) &\rightarrow \exists v D(v, v), \\ U(x, y) &\rightarrow \mu(x). \end{aligned}$$

Definition of successor predicate S. We include the following st-tgds in Σ :

$$\begin{aligned} \lambda \wedge \mu(x) \wedge \mu(y) &\rightarrow \exists u (S(x, y, u) \wedge T'(u)), \\ \lambda \wedge \mu(x) &\rightarrow \exists y \exists u (S(x, y, u) \wedge O(u)). \end{aligned}$$

As for the case of linear order L , the first st-tgd is used to indicate that for every pair of elements x, y satisfying formula μ , there is a truth value associated with the statement “ y is a successor of x according to S ”. The second st-tgd above is used to indicate that every element satisfying μ has a successor element. We note that this dependency states that even the last element of linear order L has a successor. This does not create any problems, as we do not impose any restrictions on the successor of the last element (for example, we do not say that it should be greater than the last element), and we do not use this successor when coding Turing Machine M .

To indicate that S is the successor relation associated with L , we include the following ts-tgds in Σ' :

$$\begin{aligned} S(x, y, u_1) \wedge O(u_1) \wedge U(x, u_2) \wedge Z(u_2) \wedge L(x, y, u_3) \wedge Z(u_3) &\rightarrow \exists v D(v, v), \\ S(x, y, u_1) \wedge O(u_1) \wedge U(x, u_2) \wedge Z(u_2) \wedge \\ L(x, z, u_3) \wedge O(u_3) \wedge L(z, y, u_4) \wedge O(u_4) &\rightarrow \exists v D(v, v), \\ S(x, y, z) &\rightarrow \mu(x) \wedge \mu(y). \end{aligned}$$

The first dependency above says that if y is a successor of x according to S and x is not the last element of L , then x is less than y according to L (unless there is a tuple (a, a) in D). As we mentioned above, this dependency does not impose any restrictions on the successor of the last element. The second ts-tgd says that there are no elements in between x and y if y is a successor of x and x is not the last element. The third dependency says that S is only defined for elements satisfying μ , which in particular implies that the successor of an element must satisfy μ . As for the case of predicates L , P and U , we also need to include an axiom saying that S is consistent with the equality predicate E :

$$S(x, y, u_1) \wedge O(u_1) \wedge E(x, x_1, u_2) \wedge O(u_2) \wedge \\ E(y, y_1, u_3) \wedge O(u_3) \wedge S(x_1, y_1, u_4) \wedge Z(u_4) \rightarrow \exists v D(v, v).$$

Encoding of DTM M . Now we are ready to present the dependencies that code DTM M . As for the case of predicates E , L , P , U and S , we include an extra argument in predicates H , T_0 , T_1 , T_B and S_q ($q \in Q$) to indicate whether a particular tuple is or is not in these predicates. Thus, for example, if u is an element of O (and thus represents value *true*), then $H(x, y, u)$ says that the head of M is in position y at time x , $T_0(x, y, u)$ says that the cell of the tape of M in position y has symbol 0 at time x , and likewise for symbols 1 and B, and $S_q(x, u)$ says that M is in state q at time x .

First, we include the following st-tgds in Σ :

$$\begin{aligned} \lambda \wedge \mu(x) \wedge \mu(y) &\rightarrow \exists u (H(x, y, u) \wedge T'(u)) \\ \lambda \wedge \mu(x) \wedge \mu(y) &\rightarrow \exists u (T_a(x, y, u) \wedge T'(u)) && \text{for } a \in \{0, 1, B\} \\ \lambda \wedge \mu(x) &\rightarrow \exists u (S_q(x, u) \wedge T'(u)) && \text{for } q \in Q \end{aligned}$$

We note that the first st-tgd says that for every pair of elements x, y satisfying formula μ , there is a truth value associated with the statement “the head of M is in position y at time x ”. We also observe that the second st-tgd is defined for every $a \in \{0, 1, B\}$, while the last one is defined for every $q \in Q$.

Second, we include the following ts-tgds in Σ' :

$$\begin{aligned} H(x, y, z) &\rightarrow \mu(x) \wedge \mu(y) \\ T_a(x, y, z) &\rightarrow \mu(x) \wedge \mu(y) && \text{for } a \in \{0, 1, B\}, \\ S_q(x, y) &\rightarrow \mu(x) && \text{for } q \in Q. \end{aligned}$$

These dependencies state that H , T_0 , T_1 , T_B and S_q ($q \in Q$) are only defined for elements satisfying μ .

Third, we include ts-tgds in Σ' saying that predicates H , T_0 , T_1 , T_B and S_q ($q \in Q$) are consistent with the equality predicate E . Since all these dependencies are similarly defined, we only include here the ts-tgd for predicate H :

$$H(x, y, u_1) \wedge O(u_1) \wedge E(x, x_1, u_2) \wedge O(u_2) \wedge \\ E(y, y_1, u_3) \wedge O(u_3) \wedge H(x_1, y_1, u_4) \wedge Z(u_4) \rightarrow \exists v D(v, v).$$

Notice that this ts-tgd says that unless there is a tuple (a, a) in D , if x is equal to x_1 and y is equal to y_1 according to E , then, according to H , the head of M is in position y at time x if and only if the head of M is in position y_1 at time x_1 .

Fourth, to state that each cell of the tape of M contains exactly one symbol at each moment, we include a ts-tgd in Σ' saying that it could not be the case that the cell of the

$$\begin{aligned}
& S_q(x, u_1) \wedge O(u_1) \wedge H(x, y, u_2) \wedge O(u_2) \wedge T_a(x, y, u_3) \wedge O(u_3) \wedge \\
& \quad U(x, u_4) \wedge Z(u_4) \wedge S(x, u, u_5) \wedge O(u_5) \wedge S_{q'}(u, u_6) \wedge Z(u_6) \rightarrow \exists v D(v, v), \\
& S_q(x, u_1) \wedge O(u_1) \wedge H(x, y, u_2) \wedge O(u_2) \wedge T_a(x, y, u_3) \wedge \\
& \quad O(u_3) \wedge U(x, u_4) \wedge Z(u_4) \wedge S(x, u, u_5) \wedge O(u_5) \wedge \\
& \quad \quad S(w, y, u_6) \wedge O(u_6) \wedge H(u, w, u_7) \wedge Z(u_7) \rightarrow \exists v D(v, v), \\
& S_q(x, u_1) \wedge O(u_1) \wedge H(x, y, u_2) \wedge O(u_2) \wedge T_a(x, y, u_3) \wedge O(u_3) \wedge \\
& \quad U(x, u_4) \wedge Z(u_4) \wedge S(x, u, u_5) \wedge O(u_5) \wedge T_b(u, y, u_6) \wedge Z(u_6) \rightarrow \exists v D(v, v).
\end{aligned}$$

Fig. 2. Target-to-source dependencies coding transition $\delta(q, a) = (q', b, \perp)$.

tape of M in position y at time x contains neither symbol 0 nor 1 nor \mathbb{B} (unless there is a tuple (a, a) in D):

$$T_0(x, y, u_1) \wedge Z(u_1) \wedge T_1(x, y, u_2) \wedge Z(u_2) \wedge T_{\mathbb{B}}(x, y, u_3) \wedge Z(u_3) \rightarrow \exists v D(v, v).$$

Furthermore, for every pair of distinct symbols a and b in $\{0, 1, \mathbb{B}\}$, we include a ts-tgd in Σ' saying that it could not be the case that the cell of the tape of M in position y at time x contains both a and b :

$$T_a(x, y, u_1) \wedge O(u_1) \wedge T_b(x, y, u_2) \wedge O(u_2) \rightarrow \exists v D(v, v).$$

Fifth, we include the following ts-tgd in Σ' stating that the head of M is in at most one position at each moment:

$$H(x, y_1, u_1) \wedge O(u_1) \wedge H(x, y_2, u_2) \wedge O(u_2) \wedge E(y_1, y_2, u_3) \wedge Z(u_3) \rightarrow \exists v D(v, v).$$

We observe that the dependency above together with the dependencies defining transition function δ (defined below) enforce that the head of M is in exactly one position at each moment. To state that M is in exactly one state at each moment, we include a ts-tgd in Σ' saying that it could not be the case that M is not in any state $q \in Q$ at time x :

$$\left(\bigwedge_{q \in Q} (S_q(x, u^q) \wedge Z(u^q)) \right) \rightarrow \exists v D(v, v).$$

Furthermore, for every pair of distinct states q_1 and q_2 in Q , we include a ts-tgd in Σ' saying that it could not be the case that M is in both states q_1 and q_2 (unless there is a tuple (a, a) in D):

$$S_{q_1}(x, u_1) \wedge O(u_1) \wedge S_{q_2}(x, u_2) \wedge O(u_2) \rightarrow \exists v D(v, v).$$

Sixth, we include the following ts-tgds in Σ' to define the initial configuration: the state of M is q_0 , the head of M is in the first position of the tape of M and each position of this tape contains the blank symbol.

$$\begin{aligned}
& P(x, u_1) \wedge O(u_1) \wedge S_{q_0}(x, u_2) \wedge Z(u_2) \rightarrow \exists v D(v, v), \\
& P(x, u_1) \wedge O(u_1) \wedge H(x, x, u_2) \wedge Z(u_2) \rightarrow \exists v D(v, v), \\
& P(x, u_1) \wedge O(u_1) \wedge T_{\mathbb{B}}(x, y, u_2) \wedge Z(u_2) \rightarrow \exists v D(v, v).
\end{aligned}$$

Finally, we include some ts-tgds in Σ' to code the transition function of M . Let $(q, a) \in (Q \setminus \{q_f\}) \times \Gamma$. If $\delta(q, a) = (q', b, \perp)$, then we include in Σ' the ts-tgds shown in Figure

2, and if $\delta(q, a) = (q', b, \mathbb{R})$, then we include similar ts-tgds, but where the head of M is moved to the right. Moreover, for every $a \in \{0, 1, \mathbb{B}\}$ we include the following “frame axiom” in Σ' :

$$H(x, y, u_1) \wedge Z(u_1) \wedge T_a(x, y, u_2) \wedge O(u_2) \wedge \\ U(x, u_3) \wedge Z(u_3) \wedge S(x, u, u_4) \wedge O(u_4) \wedge T_a(u, y, u_5) \wedge Z(u_5) \rightarrow \exists v D(v, v),$$

which says that unless there exists a tuple (a, a) in D , if the head of M is not in position y at time x , then the symbol in this position is the same at time u , where u is the successor of x .

Accepting condition for DTM M . To conclude the reduction, we include in Σ' the following ts-tgd:

$$S_{q_f}(x, y) \wedge O(y) \rightarrow \exists v D(v, v).$$

This dependency says that if at time x the state of the DTM M is q_f (the final state), then there exists an element a such that (a, a) is in D .

We have concluded the definitions of Σ and Σ' . We now proceed to prove that M accepts the empty string if and only if \mathcal{M}' is not a recovery of \mathcal{M} . But before we need to prove some intermediate lemmas.

LEMMA C.1. *Let I be an instance of \mathbf{S} . If $\{(a, b) \in D^I \mid a \in T^I \text{ and } b \in T^I\} = \emptyset$, then $(I, I) \in \mathcal{M} \circ \mathcal{M}'$.*

PROOF. Assume that $\{(a, b) \in D^I \mid a \in T^I \text{ and } b \in T^I\} = \emptyset$. Then I does not satisfy formula $\lambda = \exists x \exists y (D(x, y) \wedge T(x) \wedge T(y))$. Thus, given that λ is in the left-hand side of every st-tgd in Σ except for the axioms $D(x, y) \rightarrow D'(x, y)$ and $T(x) \rightarrow T'(x)$, we conclude that the following instance J of \mathbf{T} is a solution for I under \mathcal{M} : $D'^J = D^I$, $T'^J = T^I$ and $X^J = \emptyset$, for every $X \in \mathbf{T} \setminus \{D', T'\}$. By simply inspecting the set Σ' , it is possible to conclude that $(J, I) \models \Sigma'$ (since $X^J = \emptyset$ for every $X \in \mathbf{T} \setminus \{D', T'\}$, we only need to show that (J, I) satisfies dependencies $D'(x, y) \rightarrow D(x, y)$ and $T'(x) \rightarrow T(x)$, which is clearly the case). Thus, we have that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$. \square

LEMMA C.2. *Let I be an instance of \mathbf{S} . If $\{a \in \text{dom}(D^I) \mid I \models \mu(a)\} = \emptyset$, then $(I, I) \in \mathcal{M} \circ \mathcal{M}'$.*

PROOF. Analogous to the proof of the Lemma C.2. \square

LEMMA C.3. *Let I be an instance of \mathbf{S} . If there is a tuple $(a, a) \in D^I$, then $(I, I) \in \mathcal{M} \circ \mathcal{M}'$.*

PROOF. Assume that there is a tuple $(a, a) \in D^I$. Furthermore, assume that for $(a_0, b_0) \in D^I$, we have that $a_0 \in T^I$ and $b_0 \in T^I$ (if there is no such a tuple in D^I , then by Lemma C.1 we conclude that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$). Let J be a solution for I such that $D'^J = D^I$, $T'^J = T^I$, $Z^J = \{a_0\}$ and $O^J = \{b_0\}$. Given that there is a tuple $(a, a) \in D^I$, we have that (J, I) satisfies every dependency in Σ' that has formula $\exists v D(v, v)$ as its conclusion. Thus, to prove that (J, I) satisfies Σ' , we only need to show that this instance satisfies ts-tgds: $D'(x, y) \rightarrow D(x, y)$, $T'(x) \rightarrow T(x)$, $Z(x) \rightarrow T(x)$, $O(x) \rightarrow T(x)$ and $Z(x) \wedge O(y) \rightarrow D(x, y)$, which is clearly the case. We conclude that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$. \square

We have all the necessary ingredients to prove that M accepts the empty string if and only if \mathcal{M}' is not a recovery of \mathcal{M} .

PROOF OF THEOREM 9.4. (\Rightarrow) Assume that M accepts the empty string, that is, assume that from the empty tape, M reaches the final state q_f in k steps, where $k \geq 2$ since the initial state of M is different from q_f . Then define I_M as the following instance of \mathbf{S} :

$$\begin{aligned} D^{I_M} &= \{(n, m) \mid n, m \in \{1, \dots, k\} \text{ and } n \neq m\} \cup \{(f, t)\}, \\ T^{I_M} &= \{f, t\}. \end{aligned}$$

Next we show that I_M is not a solution for I_M under $\mathcal{M} \circ \mathcal{M}'$. For the sake of contradiction, assume $(I_M, I_M) \in \mathcal{M} \circ \mathcal{M}'$. Then there exists an instance J of \mathbf{T} such that $(I_M, J) \models \Sigma$ and $(J, I_M) \models \Sigma'$. Since (I_M, J) satisfies $D(x, y) \rightarrow D'(x, y)$ and (J, I_M) satisfies $D'(x, y) \rightarrow D(x, y)$, we conclude that $D'^J = D^{I_M}$, and given that (I_M, J) satisfies $T(x) \rightarrow T'(x)$ and (J, I_M) satisfies $T'(x) \rightarrow T(x)$, we conclude that $T'^J = T^{I_M}$. Moreover, given that $(f, t) \in D^{I_M}$ and $T^{I_M} = \{f, t\}$, we have that I_M satisfies λ . Thus, given that (I, J) satisfies $\lambda \rightarrow \exists u Z(u) \wedge \exists v O(v)$ and (J, I_M) satisfies $Z(x) \rightarrow T(x)$, $O(x) \rightarrow T(x)$ and $Z(x) \wedge O(y) \rightarrow D(x, y)$, we conclude that $Z^J = \{f\}$ and $O^J = \{t\}$.

Since $I_M \not\models \exists v D(v, v)$ and $(J, I_M) \models \Sigma'$, we have that (J, I_M) does not satisfy the left-hand side of any dependency in Σ' having $\exists v D(v, v)$ in its right-hand side. Furthermore, given that $k \geq 2$, we have that $\{a \in \text{dom}(D^{I_M}) \mid I_M \models \mu(a)\}$ is not empty. Thus, by considering the dependencies in Σ and Σ' that define predicates $E, L, P, U, S, H, T_0, T_1, T_B$ and S_q ($q \in Q$), we conclude that these predicates encode k steps of the run of M on the empty string. Given that M accepts the empty string, we have that $(n, t) \in S_{q_f}^J$ for some $n \in \{1, \dots, k\}$ (note that linear order L does not necessarily coincide with the usual linear order $1 < 2 < \dots < k$). Thus, given that $t \in O^J$ and (J, I_M) satisfies dependency $S_{q_f}(x, y) \wedge O(y) \rightarrow \exists v D(v, v)$, we have that there is a tuple (a, a) in D^{I_M} , which contradicts the definition of instance I_M .

(\Leftarrow) Assume that M does not accept the empty string. Thus, given that we assume that the transition function δ of M is a total function with domain $(Q \setminus \{q_f\}) \times \{0, 1, \mathbb{B}\}$, we have that for every $k \geq 1$, DTM M reaches some state $q \in (Q \setminus \{q_f\})$ in k steps from the initial empty tape. We use this fact to show that \mathcal{M}' is a recovery of \mathcal{M} .

Let I be an instance of \mathbf{S} . We need to show that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$. If $\{(a, b) \in D^I \mid a \in T^I \text{ and } b \in T^I\} = \emptyset$ or $\{a \in \text{dom}(D^I) \mid I \models \mu(a)\} = \emptyset$ or there is a tuple $(a, a) \in D^I$, then $(I, I) \in \mathcal{M} \circ \mathcal{M}'$ by Lemmas C.1, C.2 and C.3. Thus, we assume that I does not satisfy any of these conditions, and without loss of generality, we assume that $\text{dom}(D^I) = \{0, \dots, n\}$ ($n \geq 1$), $(0, 1) \in D^I$, $0 \in T^I$ and $1 \in T^I$.

To prove that I is a solution for I under $\mathcal{M} \circ \mathcal{M}'$, we construct a solution J for I under \mathcal{M} such that $(J, I) \models \Sigma'$. More precisely, we define J as follows: $D'^J = D^I$, $T'^J = T^I$,

$Z^J = \{0\}, O^J = \{1\}$ and

$$E^J := \{(i, i, 1) \mid i \in \{0, \dots, n\}\} \cup \{(i, j, 0) \mid i, j \in \{0, \dots, n\} \text{ and } i \neq j\},$$

$$L^J := \{(i, j, 1) \mid i, j \in \{0, \dots, n\} \text{ and } i < j\} \cup \{(i, j, 0) \mid i, j \in \{0, \dots, n\} \text{ and } i \geq j\},$$

$$S^J := \{(i, i + 1, 1) \mid i \in \{0, \dots, n - 1\}\} \cup$$

$$\{(n, 1, 1)\} \cup \{(i, j, 0) \mid i, j \in \{0, \dots, n\}, j \neq i + 1 \text{ and } (i, j) \neq (n, 1)\},$$

$$P^J := \{(0, 1)\} \cup \{(i, 0) \mid i \in \{1, \dots, n\}\},$$

$$U^J := \{(n, 1)\} \cup \{(i, 0) \mid i \in \{0, \dots, n - 1\}\}.$$

Furthermore, relations H^J, T_0^J, T_1^J, T_B^J and S_q^J ($q \in Q$) are defined in such a way that they represent $n + 1$ steps of the run of M on the empty string. Since M does not accept the empty string, we conclude that $(i, 1) \notin S_{q_f}^J$ for every $i \in \{0, \dots, n\}$ and, therefore, (J, I) trivially satisfies dependency $S_{q_f}(x, y) \wedge O(y) \rightarrow \exists v D(v, v)$ since $O^J = \{1\}$. Furthermore, (J, I) satisfies all the other ts-tgds in Σ' having $\exists v D(v, v)$ in its right-hand side, since H^J, T_0^J, T_1^J, T_B^J and S_q^J ($q \in Q$) code $n + 1$ steps of the run of M on the empty string. Thus, to prove that $(J, I) \models \Sigma'$, we only need to show that this instance satisfies dependencies $D'(x, y) \rightarrow D(x, y), T'(x) \rightarrow T(x), Z(x) \rightarrow T(x), O(x) \rightarrow T(x)$ and $Z(x) \wedge O(y) \rightarrow D(x, y)$. It is straightforward to prove that (J, I) satisfies the previous ts-tgds. Thus, we conclude that $(J, I) \models \Sigma'$ and, hence, $(I, I) \in \mathcal{M} \circ \mathcal{M}'$. This concludes the proof of the theorem. \square

C.4 Proof of Corollary 9.5

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and $\mathcal{M}' = (\mathbf{T}, \mathbf{S}, \Sigma')$ be the mappings used in the proof of Theorem 9.4. Recall that Σ is a set of st-tgds from \mathbf{S} to \mathbf{T} and Σ' is a set of ts-tgds from \mathbf{T} to \mathbf{S} , where

$$\mathbf{S} := \{D(\cdot, \cdot), T(\cdot)\},$$

$$\mathbf{T} := \{D'(\cdot, \cdot), T'(\cdot), Z(\cdot), O(\cdot), E(\cdot, \cdot, \cdot), L(\cdot, \cdot, \cdot), P(\cdot, \cdot), U(\cdot, \cdot), S(\cdot, \cdot, \cdot), H(\cdot, \cdot, \cdot), \\ T_0(\cdot, \cdot, \cdot), T_1(\cdot, \cdot, \cdot), T_B(\cdot, \cdot, \cdot)\} \cup \{S_q(\cdot, \cdot) \mid q \in Q\}.$$

Given that Σ includes dependencies $D(x, y) \rightarrow D'(x, y)$ and $T(x) \rightarrow T'(x)$, and Σ' includes dependencies $D'(x, y) \rightarrow D(x, y)$ and $T'(x) \rightarrow T(x)$, we have that if $(I_1, I_2) \in \mathcal{M} \circ \mathcal{M}'$, then $I_1 \subseteq I_2$. Thus, from Lemma 9.2, we have that \mathcal{M}' is a recovery of \mathcal{M} iff \mathcal{M}' is an inverse of \mathcal{M} . Hence, from the proof of Theorem 9.4, we conclude that the problem of verifying, given schema mappings \mathcal{M}_1 and \mathcal{M}_2 specified by a set of st-tgds and a set of ts-tgds, respectively, whether \mathcal{M}_2 is an inverse of \mathcal{M}_1 is undecidable.

A careful inspection of the set of dependencies Σ reveals that mapping \mathcal{M} is invertible. In fact, ts-mapping specified by dependencies $D'(x, y) \rightarrow D(x, y)$ and $T'(x) \rightarrow T(x)$ is an inverse of \mathcal{M} . Thus, from Theorem 6.3, we conclude that \mathcal{M}' is a maximum recovery of \mathcal{M} iff \mathcal{M}' is an inverse of \mathcal{M} . Therefore, from the proof of Theorem 9.4, we have that the problem of verifying, given schema mappings \mathcal{M}_1 and \mathcal{M}_2 specified by a set of st-tgds and a set of ts-tgds, respectively, whether \mathcal{M}_2 is a maximum recovery of \mathcal{M}_1 is undecidable. Furthermore, from Proposition 3.24 in [Fagin et al. 2008], we conclude that \mathcal{M}' is an inverse of \mathcal{M} iff \mathcal{M}' is a quasi-inverse of \mathcal{M} . Hence, from the proof of Theorem 9.4, we have that the problem of verifying, given schema mappings \mathcal{M}_1 and \mathcal{M}_2 specified by a set of st-tgds and a set of ts-tgds, respectively, whether \mathcal{M}_2 is a quasi-inverse of \mathcal{M}_1

70 ·

is undecidable.