# Static Analysis of Schema-Mappings Ensuring Oblivious Termination

Bruno Marnette
Oxford University Computing Laboratory
bruno.marnette@comlab.ox.ac.uk

Floris Geerts
University of Edinburgh
fgeerts@inf.ed.ac.uk

## ABSTRACT

A schema-mapping is a high level specification of a data-exchange setting where a set of source-to-target dependencies is used to realize basic operations from source to target relations (such as copy, selection, join or union) while the target schema is subject to a set of target constraints (such as inclusion dependencies or key constraints). In this paper, we consider *strong* schema-mappings that allow for additional constraints such as *source dependencies* on the source schema and *target-to-source dependencies* from the target relations back to the source. Furthermore, strong schema-mappings may include disjunctive dependencies. We argue that this extension is desirable when the source instance is to provide both a lower and upper bound on the information that a target instance can have.

We first focus on the *implication problem* for strong schema-mappings which is to determine whether a given constraint $\delta$ is logically implied by the set $\Sigma$ of constraints (denoted by $\Sigma \models \delta$). After providing complete characterizations for this problem in terms of universal solutions (while supporting equality constraints), we introduce criteria of termination, denoted by TOC, DTOC and MTOC, that allow the efficient computation of universal solutions for standard constraints, disjunctive constraints, and when the source instance is assumed to be immutable (i.e., it is master data), respectively. We obtain decision procedures for the implication problem, provided that $\Sigma$ satisfies these termination conditions, and give the corresponding complexity bounds. As an immediate application we revisit the problems of determinacy, relative information completeness and variations thereof, all for strong schema-mappings. Indeed, by viewing them as implication problems we obtain efficient decision procedures when the relevant termination conditions are satisfied.

We then focus on the problem of deciding whether source-to-target constraints in a strong schema-mapping are already implied by the embedded (standard) schema-mapping. This problem is important if one wants to use target-to-source constraints in standard data-exchange tools. Since no such constraints are logically implied by standard schema-mappings (and hence the results established earlier are of no use), we provide an alternative semantics for implication. More specifically, we want the constraint to be satisfied by every solution corresponding to the output of a standard data-exchange tool. We consider three semantics based on universal solutions, cores and CWA-solutions, respectively. Decidability of the implication of general (resp. safe) target-to-source constraints is shown for the CWA-based semantics (resp. core-semantics).


*ICDT 2010*, March 22–25, 2010, Lausanne, Switzerland.


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

**General Terms:** Algorithms, Theory.

**Keywords:** Schema mapping, data exchange, data integration, relative completeness, determinacy.

## 1. INTRODUCTION

A schema-mapping is a high-level specification that describes the relationship between two database schemas. As schema-mapping typically consists of a tuple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ where $\mathbf{S}$ is a source schema, $\mathbf{T}$ is a target schema, $\Sigma_{st}$ is a set of *source-to-target dependencies* (tgds) that describe the relationship between the two schemas and $\Sigma_t$ is a set of *target dependencies* (tgds and egds) that describe constraints on the target instance. Since schema-mappings are crucial for data interoperability tasks (see e.g., [14]), an extensive investigation into the foundations of schema-mappings has been carried out in recent years. In particular, one main line of investigations has focused on the computation of *solutions* of schema-mappings [6, 7, 11, 23]. Here, given $\mathcal{M}$ and a source instance $I$ of $\mathbf{S}$, a target instance $J$ of $\mathbf{T}$ is a solution for $\mathcal{M}$ and $I$ if $(I, J) \models \Sigma$ for $\Sigma = \Sigma_{st} \cup \Sigma_t$. These solutions play a prominent role throughout this paper.

In spite of its success, schema-mappings are limited in the sense that constraints in $\Sigma$ only provide a *lower bound* on the set of "true" facts in the solutions. Similarly, a solution is understood as a *sound view* of the schema $\mathbf{T}$, stating that some facts must hold, but without providing any upper-bound.

EXAMPLE 1.1. Consider the following schema-mapping with source schema

$$\{\mathsf{Emp}(\text{name}, \text{salary}, \text{area}); \mathsf{Client}(\text{name}, \text{phone}, \text{country})\}$$

which stores employees and clients in a world-wide company. The UK branch of the company has target schema

$$\{\mathsf{Emp}_{\mathsf{UK}}(\text{name}, \text{salary})); \mathsf{Client}_{\mathsf{UK}}(\text{name}, \text{phone})\}.$$

The following set $\Sigma_{st}$ of source-to-target dependencies can be used to (1) export the list of employees already assigned to the UK while authorizing the branch to adapt their salary and (2) export the list of clients located in the UK:

$$\mathsf{Emp}(e, s, UK) \rightarrow \exists s' \ \mathsf{Emp}_{\mathsf{UK}}(e, s')$$
$$\mathsf{Client}(n, p, UK) \rightarrow \mathsf{Client}_{\mathsf{UK}}(n, p)$$

If we consider the source instance $I$ below (left) then a solution $J$ for $\mathcal{M}$ and $I$ is given below (right).

| $I$ | $J$ |
|---|---|
| $\mathsf{Emp}(\text{alice}, 3000, UK)$ | |
| $\mathsf{Emp}(\text{bob}, 2500, USA)$ | |
| $\mathsf{Emp}(\text{cedric}, 3500, UK)$ | $\mathsf{Emp}_{\mathsf{UK}}(\text{alice}, \#_a)$ |
| $\mathsf{Client}(\text{denise}, 01234098, UK)$ | $\mathsf{Emp}_{\mathsf{UK}}(\text{cedric}, \#_c)$ |
| $\mathsf{Client}(\text{edward}, 01899785, AU)$ | $\mathsf{Client}_{\mathsf{UK}}(\text{denise}, 01234098)$ |
| $\mathsf{Client}(\text{fred}, 06947647, US)$ | |

Any other solutions is obtained by instantiating the *nulls* $\#_a$ and $\#_c$ in $J$ arbitrarily. Hence, no upper bound on neither those values nor the cardinality of solutions is provided. □

In many practical settings, however, it is desirable to also provide an "upper bound" on the solutions of a schema-mapping.

Example 1.2. Continuing the previous example, suppose that a list $\mathsf{Sal}$ of possible salaries for employees in the UK branch is available. One could then enforce the nulls to take values from this list, provided *disjunctive target* constraints are allowed. Indeed, adding

$$\forall xy \ \mathsf{Emp}_{\mathsf{UK}}(x, y) \rightarrow \bigvee_{s \in \mathsf{Sal}} y = s$$

to $\Sigma_t$ would provide the desired upper bound. Similarly, suppose that one wants to encode that every employee of the UK branch has to be registered in the main company, or equivalently, that the branch cannot hire a new employee without notifying the main company (which would lead to updating the source instance). For this, *target-to-source dependencies* are needed. Indeed, the dependency

$$\delta: \quad \mathsf{Emp}_{\mathsf{UK}}(e, s) \rightarrow \exists s', l' \ \mathsf{Emp}(e, s', l')$$

suffices for this. Furthermore, one needs to distinguish between the case when the source instance $I$ is assumed to be immutable (i.e., $I$ is so-called *master data* using the terminology from [8]), and when $I$ can be updated. In the former case, $J$ can be updated to $J'$ below (where the salaries have been fixed and where clients have been added) but cannot be updated to $J''$ below (where a new employee has been added). When $I$ can be updated, both solutions are feasible.

| $J'$ | $J''$ |
|---|---|
| $\mathsf{Emp}_{\mathsf{UK}}(\text{alice}, 3000)$ | $\mathsf{Emp}_{\mathsf{UK}}(\text{alice}, 3000)$ |
| $\mathsf{Emp}_{\mathsf{UK}}(\text{cedric}, 3500)$ | $\mathsf{Emp}_{\mathsf{UK}}(\text{cedric}, 3500)$ |
| $\mathsf{Client}_{\mathsf{UK}}(\text{denise}, 01234098)$ | $\mathsf{Emp}_{\mathsf{UK}}(\text{heather}, 3500)$ |
| $\mathsf{Client}_{\mathsf{UK}}(\text{georg}, 01993707)$ | $\mathsf{Client}_{\mathsf{UK}}(\text{denise}, 01234098)$ |
| $\mathsf{Client}_{\mathsf{UK}}(\text{isabel}, 01993707)$ | $\mathsf{Client}_{\mathsf{UK}}(\text{georg}, 01993707)$ □ |

Motivated by the previous examples, it therefore seems natural to extend schema-mappings to *strong* schema-mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ where $\Sigma$ now consists of $\Sigma_t$ (target dependencies), $\Sigma_{st}$ (source-to-target dependencies), $\Sigma_s$ (source dependencies) and finally $\Sigma_{ts}$ (target-to-source dependencies).

This more general setting, has been previously studied by Deutsch et al [4]. In fact, it is shown there that most concepts and techniques from standard schema-mappings generalize to the strong schema-mapping setting. In particular, they studied the *implication problem* which is to decide, given a strong schema-mapping $\mathcal{M}$ and a dependency $\delta$ over $\mathbf{S} \cup \mathbf{T}$, whether $\Sigma$ *logically implies* $\delta$, denoted by $\Sigma \models \delta$. In other words, it is to decide whether for any source instance $I$ of $\mathbf{S}$ and *any* solution $J$ for $\mathcal{M}$ and $I$, $(I, J) \models \delta$. It is shown in [4] that when $\Sigma$ consists of *tgds only*, *universal* solutions can be used to determine $\Sigma \models \delta$, provided that these solutions can be computed. However, no termination conditions of the chase procedure (that returns universal solutions) are established in [4] when $\Sigma$ consists of disjunctive constraints or when the source instance is assumed to be immutable.

We therefore believe that the implication problem for (strong) schema-mappings needs revisiting and this is indeed the first main focus of this paper. Our first contributions (Section 3) are the following: Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a strong schema-mapping and $\delta$ a constraint over $\mathbf{S} \cup \mathbf{T}$.

- We provide characterizations for when $\Sigma \models \delta$ holds in terms of universal solutions when $\Sigma$ possibly contains egds and/or disjunctive constraints, hereby extending the work of [4].
- Termination conditions of the *oblivious chase* [17] are given for standard constraints, disjunctive constraints and for the case when the source instance is assumed to be immutable (master data). The corresponding classes of strong schema-mappings are denoted by $\mathsf{TOC}$, $\mathsf{DTOC}$ and $\mathsf{MTOC}$, respectively.
- Finally, complexity bounds for $\Sigma \models \delta$ are established under the assumption that the constraints belong to $\mathsf{TOC}$, $\mathsf{DTOC}$ or $\mathsf{MTOC}$. In particular, the implication problem is in NP when $\mathcal{M} \in \mathsf{TOC}$, and in $\Pi_2^p$ when $\mathcal{M} \in \mathsf{DTOC}$ or $\mathcal{M} \in \mathsf{MTOC}$. The complexity bounds carry over to the class of weakly acyclic [6] or super-weakly acyclic [17] schema-mappings.

As an immediate application of these results, we revisit the problems of relative information completeness [8] and determinacy [21, 19] in the strong schema-mapping setting in Section 4. Indeed, we show that the two problems can be viewed as an implication problem and hence, by leveraging the results in Section 3, we establish complexity bounds for both these problems under some termination conditions.

The second main focus of this paper addresses the important question whether the target-to-source constraints in a strong schema-mapping are already implied by the embedded (standard) schema-mapping. More precisely, given a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where as before $\Sigma$ consists of $\Sigma_{st}$, $\Sigma_{ts}$, $\Sigma_s$ and $\Sigma_t$, does $\mathcal{M}' = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$, already imply the constraints in $\Sigma_{ts}$? This question is important for the reason that current schema-mapping management systems (such as e.g., [12]) do not support target-

to-source constraints. Consequently, only target-to-source constraints can be included that are "harmless" (i.e., already implied by standard constraints).

Note that the results in Section 3 are of no use for this problem. Indeed, standard schema-mappings can only logically entail source-to-target or target constraints. We are therefore faced with the problem of properly defining the semantics of entailment for target-to-source constraints in the (standard) schema-mapping setting, while at the same time ensuring decidability of the implication problem for large classes of constraints. Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a strong schema-mapping and assume that $\Sigma_{ts} = \{\delta\}$. We explore under which semantics of solutions (denoted by $\mathsf{Sem}$) we can be sure that, for all source instances $I$ and every $J$ in $\mathsf{Sem}(\mathcal{M}, I)$ we have $(I, J) \models \delta$.

Our contributions (Section 5) consists of the following:

- We show that the semantics based on *universal solutions* ($\mathsf{USol}$) is not restrictive enough while the semantics based on *cores of universal solutions* ($\mathsf{Core}$) is on the contrary too restrictive and leads very easily to undecidability, even for weakly-acyclic schema-mappings.
- We propose an intermediate semantics, which in the case of tgds only, coincides with the notion of *closed-world assumption* semantics [15, 13].
- We show that when the embedded standard schema-mapping $\mathcal{M}' \in \mathsf{TOC}$, the CWA-based implication problem is decidable for arbitrary target-to-source dependencies. The core-based implication problem is decidable for *safe* target-to-source constraints only.

**Organization.** In Section 2 we define schema-mappings and universal solutions. We define strong schema-mappings and study its corresponding implication problem in Section 3, and show as an application its use for the problems of completeness and determinacy in Section 4. The implication problem for target-to-source constraints by the embedded standard schema-mapping is studied in Section 5. Finally, Section 6 discusses related work.

## 2. PRELIMINARIES

**Terms.** We fix a universal schema $\mathbf{U}$ consisting of an infinite set of predicate symbols $R$ of fixed and finite arity $a_R$. We define a schema $\sigma$ as a subset of $\mathbf{U}$. We also fix an infinite set $\mathsf{Cst}$ of constants, an infinite set $\mathsf{Var}$ of variables, and a set $\mathsf{Functs}$ of function symbols $f$ of fixed and finite arity $a_f$. We define the set $\mathsf{Terms}$ of terms as the minimal set containing $\mathsf{Cst} \cup \mathsf{Var}$ and such that $f\langle t_1, \ldots, t_n\rangle$ is a term whenever $f \in \mathsf{Functs}$, $n = a_f$ and $t_1, \ldots, t_n \in \mathsf{Terms}$. We inductively define the *size* $|t|$ of a term $t$ as $|t| = 1$ if $t \in \mathsf{Cst} \cup \mathsf{Var}$ and $|t| = 1 + |t_1| + \ldots + |t_n|$ if $t = f\langle t_1, \ldots, t_n\rangle$. We let $\mathsf{Dom}$ be the set of terms in which no variables occur, and we define the set $\mathsf{Nulls}$ of nulls as the set of non-constant terms in $\mathsf{Dom}$ so that $\mathsf{Dom} = \mathsf{Cst} \uplus \mathsf{Nulls}$. We say that a null is *flat* if it is of the form $f\langle\rangle$ for some $f \in \mathsf{Functs}$ of arity zero. In this paper, we use either integers $1, 2, \ldots$ or (bold) letters $\mathbf{a}, \mathbf{b}, \ldots$ to denote constants, we use $x, y, z, u, v, \ldots$ to denote variables, and we use $\#_1, \#_2, \ldots$ to denote flat nulls.

**Instances and Morphisms.** We define a *fact* as an object $R(t_1, \ldots, t_n)$ where $R \in \mathbf{U}$, $n = a_R$ and $t_1, \ldots, t_n \in \mathsf{Dom}$. We define an *instance* as a finite or infinite set of facts. (The semantics of finite instances is discussed in Section 6.) Given an instance $I$, we let $\mathsf{Cst}(I)$ and $\mathsf{Nulls}(I)$ be the set of constants and nulls occurring in $I$ and we define the active domain of $I$ as $\mathsf{Dom}(I) = \mathsf{Cst}(I) \cup \mathsf{Nulls}(I)$. We say that an instance $I$ is *ground* iff $\mathsf{Dom}(I) = \mathsf{Cst}(I)$ and $\mathsf{Nulls}(I) = \emptyset$. We denote by $\sigma_I$ the set of predicate symbols occurring in an instance $I$ and say that $I$ is *over* a schema $\sigma$ iff $\sigma_I \subseteq \sigma$. Given a mapping $h : \mathsf{Dom}(I) \to \mathsf{Dom}$, we denote by $h(I)$ the instance containing the fact $R(h(t_1), \ldots, h(t_n))$, for each fact $R(t_1, \ldots, t_n)$ in $I$. Given two instances $I$ and $J$, an *homomorphism* $h$ from $I$ to $J$ is a mapping $h : \mathsf{Dom}(I) \to \mathsf{Dom}(J)$ such that $h(I) \subseteq J$ and $h(c) = c$ for all constant $c \in \mathsf{Cst}(I)$. We write $I \xrightarrow{\mathsf{hom}} J$ iff there exists a homomorphism from $I$ to $J$.

**Tgds and Egds.** An *atom* is an object $R(t_1, \ldots, t_n)$ where $R \in \mathbf{U}$, $n = a_R$ and $t_1, \ldots, t_n \in \mathsf{Var} \cup \mathsf{Cst}$ (note the difference with the definition of a fact given earlier). Given a set $\phi$ of atoms we let $\mathsf{Var}(\phi)$ and $\mathsf{Cst}(\phi)$ be the sets of variables and constants occurring in $\phi$. We define a *tuple-generating dependency*, or *tgd* for short, as a first-order formula $r$ of the form

$$\forall \overline{x}, \overline{y},\ \phi(\overline{x}, \overline{y}) \to \exists \overline{z}, \psi(\overline{x}, \overline{z})$$

where $\phi$ and $\psi$ are two finite sets of atoms called respectively the *body* and the *head* of $r$ and $\overline{x}$,$\overline{y}$ and $\overline{z}$ are three disjoint tuples of distinct variables such that $\{\overline{x}\} = \mathsf{Var}(\phi) \cap \mathsf{Var}(\psi)$; $\{\overline{y}\} = \mathsf{Var}(\phi) \setminus \{\overline{x}\}$ and $\{\overline{z}\} = \mathsf{Var}(\psi) \setminus \{\overline{x}\}$.
We define an *equality-generating dependency*, or *egd* for short, as a first-order formula $s$ of the form

$$\forall \overline{x}, \phi(\overline{x}) \to \alpha(\overline{x}) = \beta(\overline{x})$$

where $\phi$ is a finite set of atoms called the *body* of $s$, and $\alpha(\overline{x})$ and $\beta(\overline{x})$ are two terms consisting either of a constant or a variable of $\overline{x}$.

**Disjunctive dependencies.** We define a *disjunctive tgd* as a first-order formula of the form

$$\forall \overline{x}, \overline{y},\ \phi(\overline{x}, \overline{y}) \to (\exists \overline{z}_1, \psi_1(\overline{x}, \overline{z}_1)) \vee \cdots \vee (\exists \overline{z}_k, \psi_k(\overline{x}, \overline{z}_k))$$

where for each $i \in [1, k]$, $\forall \overline{x}, \overline{y},\ \phi(\overline{x}, \overline{y}) \to \exists \overline{z}_i, \psi(\overline{x}, \overline{z}_i)$ is a tgd. We define a *disjunctive egd* as a first-order formula of the form

$$\forall \overline{x}, \phi(\overline{x}) \to (e_{1,1} \wedge \cdots \wedge e_{1,p_1}) \vee \cdots \vee (e_{k,1} \wedge \cdots \wedge e_{k,p_k})$$

where for each $i \in [1, k]$ and each $j \in [1, p_k]$, $e_{ij}$ denotes an equality $\alpha_{i,j}(\overline{x}) = \beta_{i,j}(\overline{x})$ and $\forall \overline{x}\ \phi(\overline{x}) \to e_{ij}$ is an egd.

Given a set $\Sigma$ of (disjunctive) tgds and egds we denote by $\mathsf{Cst}(\Sigma)$ the set of constants occurring in $\Sigma$ and by $\sigma_\Sigma$ the set of predicate symbols occurring in $\Sigma$. Given an instance $I$ we write $I \models \Sigma$ iff $I$ is a model of $\Sigma$ according to the standard semantics of FO. Given two sets $\Sigma$ and $\Sigma'$ of constraints we write $\Sigma \models \Sigma'$ iff for all (possibly infinite) instance $I$ we have that $I \models \Sigma$ implies $I \models \Sigma'$. In the following sections, we often omit to write explicitly the universal quantification in front of the (disjunctive) tgds and egds.

**Schema-Mappings.** A schema-mapping is a tuple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ where: **(i)** $\mathbf{S}$ and $\mathbf{T}$ are two disjoint finite schemas respectively called *source schema* and *target schema*; **(ii)** $\Sigma_{st}$ is a finite set of tgds called *source-to-target tgds* of the form $\phi \to \psi$ for some $\phi$ over $\mathbf{S}$ and some $\psi$ over $\mathbf{T}$; and **(iii)** $\Sigma_t$ is a finite set of tgds over $\mathbf{T}$ called *target tgds* and egds over $\mathbf{T}$ called *target egds*. Given a schema-mapping $\mathcal{M}$,

a *source instance* for $\mathcal{M}$ is a ground instance over $\mathbf{S}$ (possibly infinite, with constants only) while a *target instance* for $\mathcal{M}$ is an instance over $\mathbf{T}$ (possibly infinite and with nulls).

**Universal Solutions and Cores.** Given a schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ and a source instance $I$, a target instance $J$ is a *universal solution* and we write $J \in \mathsf{USol}(\mathcal{M}, I)$ iff (i) $J$ is a solution, meaning that $I \cup J \models (\Sigma_{st} \cup \Sigma_t)$, and (ii) $J$ is a universal, meaning that for all solutions $K$ we have $J \xrightarrow{\text{hom}} K$. A universal solution $J \in \mathsf{USol}(\mathcal{M}, I)$ is a *core of universal solutions* and we write $J \in \mathsf{Core}(\mathcal{M}, I)$ iff $J$ is finite, and there exists no strict subset $J' \subset J$ such that $J' \in \mathsf{USol}(\mathcal{M}, I)$.

**Universal models.** In general, one may consider any finite set $\Sigma$ of tgds and egds (not necessarily restricted to the schema-mapping format). Similar to the notion of universal solution, and following the terminology of [4], an instance $J$ is a *universal model* of $\Sigma$ and an instance $I$ if (i) $I \xrightarrow{\text{hom}} J$; (ii) $J \models \Sigma$; and (iii) for every instance $K$, if $K \models \Sigma$ and $I \xrightarrow{\text{hom}} K$, then $J \xrightarrow{\text{hom}} K$. We denote by $\mathsf{UMod}(\Sigma, I)$ the set of universal models of $\Sigma$ and $I$.

# 3. LOGICAL ENTAILMENT

The *implication problem* for a class $\mathcal{C}$ of dependencies is the following: given a finite set $\Sigma$ of dependencies, $\Sigma \subseteq \mathcal{C}$, and a dependency $\delta \in \Sigma$, does $\Sigma \models \delta$ ? In this section, we study the implication problem for (disjunctive) tgds and egds, a problem which is also known as the problem of *logical entailment* in the context of schema-mappings. In order to capture the generality promised in the introduction, we first define a notion of *strong* schema-mappings.

DEFINITION 3.1. *We define a* strong *schema-mapping $\mathcal{M}$ as a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ where $\Sigma = \Sigma_s \uplus \Sigma_{st} \uplus \Sigma_{ts} \uplus \Sigma_t$ is a set of disjunctive tgds and egds decomposed as follows:*

- *$\Sigma_{st}$ is a set of disjunctive tgds from $\mathbf{S}$ to $\mathbf{T}$;*
- *$\Sigma_t$ is a set of disjunctive tgds and egds over $\mathbf{T}$;*
- *$\Sigma_{ts}$ is a set of disjunctive tgds from $\mathbf{T}$ to $\mathbf{S}$;*
- *$\Sigma_s$ is a set of disjunctive tgds and egds over $\mathbf{S}$.*

In this definition, $\Sigma_{st}$ corresponds to the source-to-target dependencies of a standard schema-mapping, $\Sigma_t$ and $\Sigma_s$ captures the dependencies that often come with a schema (e.g., tgds encoding inclusion dependencies, egds encoding key constraints, and disjunctive egds encoding domain constraints), and $\Sigma_{ts}$ is a set of target-to-source dependencies allowing to encode complete views (see also Section 5).

Given a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and a source instance $I$, we denote by $\mathsf{Sol}(\mathcal{M}, J)$ the set of target instances $J$ such that $(I, J) \models \Sigma$. The following observation is straightforward.

OBSERVATION 3.1. *Given a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and a formula $\delta$ over $\mathbf{S} \cup \mathbf{T}$, the following statements are equivalent:*

- $\Sigma \models \delta$
- *for every source instance $I$ and every target instance $J \in \mathsf{Sol}(\mathcal{M}, I)$ we have $(I, J) \models \delta$.*

To obtain decidability for the implication problem, we provide a complete characterization that (unlike previous approaches) supports egds. More specifically, we first recall the characterization given in [4] and show that it is a complete characterization in the case of tgds only. We then show how the notion of *substitution-free simulation* of egds introduced in [11, 17] can be used to also provide a complete characterization in the presence of egds.

## 3.1 Implication under Tgds and Egds

We first recall a well-known (partial) characterization of the implication problem for tgds and egds. The constraints in this and the following section are non-disjunctive; disjunctive constraints are considered in Section 3.3

PROPOSITION 3.1 ([4]). *Given a finite set $\Sigma$ of tgds and egds, given a tgd $\delta$ of the form $\phi(\bar{x}, \bar{y}) \rightarrow \exists z\ \psi(\bar{x}, \bar{z})$, given two tuples of fresh constants $\bar{a} \in \mathsf{Dom}^{|x|}$ and $\bar{b} \in \mathsf{Dom}^{|y|}$, and given a universal model $K \in \mathsf{UMod}(\Sigma, \phi(\bar{a}, \bar{b}))$ we have $\Sigma \models \delta$ iff $K \models \exists \bar{z}, \psi(\bar{a}, \bar{z})$.*

When $\Sigma$ consists only of tgds, it can be seen that a universal model $K$ necessarily exists (even though it might be infinite) and therefore the characterization offered by Proposition 3.1 is complete in the case without egds. More precisely, using the notations of Proposition 3.1 the following statements are all equivalent when $\Sigma$ is a set of tgds only:

- $\Sigma \models \delta$
- $\exists K \in \mathsf{UMod}(\Sigma, \phi(\bar{a}, \bar{b}))$, $K \models \exists \bar{z}, \psi(\bar{a}, \bar{z})$
- $\forall K \in \mathsf{UMod}(\Sigma, \phi(\bar{a}, \bar{b}))$ $K \models \exists \bar{z}, \psi(\bar{a}, \bar{z})$

An important observation here is that the above characterization is not complete in the presence of egds, simply because universal models may not exist. This is illustrated in the following example.

EXAMPLE 3.1. Consider $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ with $\mathbf{S} = \emptyset$, $\mathbf{T} = \{A, B, C\}$ and $\Sigma = \Sigma_t$, $\delta_1$ and $\delta_2$ defined as:

$$\Sigma_t = \left\{ \begin{array}{l} A(x,y) \rightarrow x = y \\ A(x,x) \wedge B(y) \rightarrow x = y \\ A(x,x) \wedge B(x) \rightarrow C(x) \end{array} \right\}$$

$$\delta_1 = A(x,y) \rightarrow C(x)$$
$$\delta_2 = A(x,y) \wedge B(z) \rightarrow C(y)$$

Clearly, $\Sigma \not\models \delta_1$ and $\Sigma \models \delta_2$. It is easily verified that no universal model exists in neither $\mathsf{UMod}(\Sigma, \{A(a,b)\})$ (for $\delta_1$) nor $\mathsf{UMod}(\Sigma, \{A(a,b), B(c)\})$ (for $\delta_2$). Hence, Proposition 3.1 is of no use to decide whether $\delta_1$ or $\delta_2$ are implied.

We next show how egds can be incorporated by means of a transformation process (see [11, 17]) that transforms a set $\Sigma$ consisting of tgds and egds into a set $s(\Sigma)$ consisting of tgds only.

**Simulation** $\epsilon(\Sigma)$. Given a set $\Sigma$ of tgds and egds, we let $\epsilon(\Sigma)$ be the set of tgds obtained from $\Sigma$ by: **(i)** replacing each equality atom $(x = y)$ by $E(x, y)$, where $E$ is a fresh binary predicate symbol; **(ii)** adding the tgds $E(x, y) \rightarrow E(y, x)$ and $E(x, y) \wedge E(y, z) \rightarrow E(x, z)$ to encode the symmetry and transitivity of $E$; and adding finally for all predicates $R$ occurring in $\Sigma$ the tgd $R(x_1, \ldots, x_{a_R}) \rightarrow E(x_1, x_1) \wedge \cdots \wedge E(x_{a_R}, x_{a_R})$ to encode the reflexivity of $E$.

**Singularisation.** Given a finite set of atoms $\phi$ over a schema $\sigma$ (that does not contain $E$) we define a *singularisation* of $\phi$ as a set of atoms $\phi^\sigma \cup \phi^E$ obtained, starting from $\phi^\sigma := \phi$ and $\phi^E := \emptyset$, and by repeating the following operations until no constant occur in $\phi^\sigma$ and no variable occurs more than once in $\phi^\sigma$:

- if a constant $\mathbf{c}$ occurs in $\phi^\sigma$, introduce a fresh variable $u$, replace every occurrence of $\mathbf{c}$ by $u$ in $\phi^\sigma$ and add the atom $E(u, \mathbf{c})$ to $\phi^E$;
- if a variable $x$ occurs more than once in $\phi^\sigma$, introduce a fresh variable $x'$, replace one occurrence of $x$ by $x'$ in $\phi^\sigma$ and add the atom $E(x, x')$ to $\phi^E$.

Note that there may be an exponential number of possible non-equivalent singularizations for a given set $\phi$ of atoms, but to simplify the discussion, we assume a fixed algorithm $\mathsf{sing}$ that, given a set $\phi$ of atoms, returns a singularisation $\mathsf{sing}(\phi)$ of $\phi$. Given a set $\Sigma$ of tgds and egds, we then define the *substitution-free simulation* $s(\Sigma)$ of $\Sigma$ as the set of tgds obtained from $\epsilon(\Sigma)$ by replacing each tgd $(\phi \rightarrow \psi)$ in $\epsilon(\Sigma)$ by $(\mathsf{sing}(\phi) \rightarrow \psi)$.

Example 3.2. Consider again the target constraints of Example 3.1. The simulation $\epsilon(\Sigma)$ consists of

$$\epsilon(\Sigma) = \left\{ \begin{array}{ll} A(x,y) \rightarrow E(x,y) & (r_1) \\ A(x,x) \wedge B(y) \rightarrow E(x,y) & (r_2) \\ A(x,x) \wedge B(x) \rightarrow C(x) & (r_3) \\ E(x,y) \wedge E(y,z) \rightarrow E(x,z) & (\text{transitivity}) \\ E(x,y) \rightarrow E(y,z) & (\text{symmetry}) \\ A(x,y) \rightarrow E(x,x) \wedge E(y,y) & (\text{reflexivity}) \\ B(y) \rightarrow E(y,y) & (\text{reflexivity}) \\ C(y) \rightarrow E(y,y) & (\text{reflexivity}) \end{array} \right\}$$

In this case, the substitution-free simulation $s(\Sigma)$ only replaces $r_2$ and $r_3$ by

$$\begin{array}{ll} r_2': & A(x,x') \wedge B(y) \wedge E(x,x') \rightarrow E(x,y) \\ r_3': & A(x,x') \wedge B(x'') \wedge E(x,x') \wedge E(x',x'') \rightarrow C(x) \end{array} \quad \square$$

We are now almost ready to provide a complete characterization for $\Sigma \models \delta$ in the presence of egds. Given an instance $I$ where the predicate symbol $E$ is used to encode equality constraints, we define $I^+$ as the instance obtained by applying to $I$ the substitution axioms of equality, that is, $I^+$ is the set of atoms $R(t_1, \ldots, t_n)$ such that $I$ contains $\{R(t_1', \ldots, t_n'); E(t_1, t_1'); \ldots; E(t_n, t_n')\}$ for some terms $t_1', \ldots, t_n' \in \mathsf{Dom}(I)$. Given a tgd $\delta$ of the form $\phi(\bar{x}, \bar{y}) \rightarrow \exists z\ \psi(\bar{x}, \bar{z})$, and considering two tuples of fresh constants $\overline{a} \in \mathsf{Dom}^{|x|}$ and $\overline{b} \in \mathsf{Dom}^{|y|}$ we let $B_\delta = \phi(\overline{a}, \overline{b})$ and $H_\delta = \exists \overline{z}, \phi(\overline{a}, \overline{z})$. Given an egd $\delta$ of the form $\phi(\overline{x}) \rightarrow \alpha(\overline{x}) = \beta(\overline{x})$ and considering a tuple $\overline{a} \in \mathsf{Dom}^{|x|}$ of fresh constants, we let $B_\delta = \phi(\overline{a})$ and $H_\delta = E(\alpha(\overline{x}), \beta(\overline{x}))$.

Proposition 3.2. *Given a finite set $\Sigma$ of tgds and egds, given a tgd or egd $\delta$, and given a universal model $K \in \mathsf{UMod}(s(\Sigma), B_\delta)$, we have $\Sigma \models \delta$ iff one of the following statements hold:*

- $K^+ \models H_\delta$, *or*
- $\exists\, c, c' \in \mathsf{Cst}(\Sigma \cup \delta),\ c \neq c' \ \wedge\ E(c, c') \in K^+$.

The following examples shed light on the two conditions in Proposition 3.2.

Example 3.3. Consider $s(\Sigma)$ given in Example 3.2 and $\delta_2$ from Example 3.1. The following instance $K$ is a universal model for $s(\Sigma)$ and $B_{\delta_2} = \{A(a,b), B(c)\}$:

$$\begin{array}{rl} K = \{ & A(a,b); B(c); E(a,b); E(b,a); E(b,c); \\ & E(c,b); E(a,c); E(c,a); C(a)\ \} \end{array}$$

Since $C(b) \in K^+$, we have $\Sigma \models \delta_2$. $\square$

Example 3.4. Consider $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ with $\mathbf{S} = \emptyset$, $\mathbf{T} = \{A, B\}$ and $\Sigma = \{A(x) \rightarrow x = 1\}$ and $\delta = A(0) \rightarrow B(0)$. Here $s(\Sigma)$ contains $A(x) \rightarrow E(x, 1)$ and a universal solution $K$ for $s(\Sigma)$ and $B_\delta = \{A(0)\}$ clearly exists. However, $K$ must contains $E(0,1)$ which means that two different constants must be identified. Since these constants originate from $\Sigma$ and $\delta$, this not allowed. Note that this precisely corresponds to the fact that $\Sigma \wedge \{A(0)\}$ is unsatisfiable. Hence, in this case, $\delta$ is vacuously implied by $\Sigma$, a situation that did not occur in the tgd case only. $\square$

## 3.2 Oblivious Termination

In order to obtain decidability for the implication problem, Proposition 3.2 tells that it suffices to identify classes of dependencies allowing (after substitution-free simulation) the computation of universal models. In this section, we first recall the notion of *oblivious termination* introduced in [17] for tgds and egds before extending this notion to disjunctive dependencies in the next section.

**Skolemization $P_\Sigma$.** Given a set of tgds $\Sigma$ we define the *Skolemization* $P_\Sigma$ of $\Sigma$ as the logic program obtained by replacing each tgd of the form

$$\phi(\overline{x}, \overline{y}) \rightarrow \exists z_1, \ldots, z_n\ \psi(\overline{x}, z_1, \ldots, z_n)$$

by a rule of the form

$$\phi(\overline{x}, \overline{y}) \rightarrow \psi(\overline{x}, f_1\langle\overline{x}\rangle, \ldots, f_n\langle\overline{x}\rangle)$$

where each $f_i$ is a fresh function symbol in $\mathsf{Functs}$ of arity $|\overline{x}|$. Given a logic program $P_\Sigma$ and an instance $I$ we denote by $P_\Sigma(I)$ the minimal Herbrand model of $P_\Sigma(I)$, or equivalently, the least-fixed point of $I$ by $P_\Sigma$. We say that $P_\Sigma$ *terminates* on $I$ iff $P_\Sigma(I)$ is finite.

Proposition 3.3 ([17]). *For any instance $I$ and set $\Sigma$ of tgds we have that $P_\Sigma(I) \in \mathsf{UMod}(\Sigma, I)$.*

**TOC.** Given a finite set of tgds $\Sigma$, we say that $\Sigma$ *ensures oblivious termination* and write $\Sigma \in \mathsf{TOC}$ iff $P_\Sigma$ terminates on every finite ground instance. Given a set $\Sigma$ of tgds and egds, we write $\Sigma \in \mathsf{TOC}^e$ iff $s(\Sigma) \in \mathsf{TOC}$. It can easily be observed that a set $\Sigma$ of tgds (without egds) is in $\mathsf{TOC}$ iff $s(\Sigma)$ is in $\mathsf{TOC}^e$ and we can therefore drop the "e" in the notation $\mathsf{TOC}^e$ without causing any ambiguity. Note that the class $\mathsf{TOC}$ was originally defined in [17] for schema-mappings $(\mathbf{S}, \mathbf{T}, \Sigma)$ where $\Sigma = \Sigma_{st} \cup \Sigma_t$ and was only requiring the termination of $P_\Sigma$ (or $P_{s(\Sigma)}$) for every finite ground source instance $I$. This requirement is however not sufficient in the context of logical entailment because $P_\Sigma$ may terminate on every finite source instance (over $\mathbf{S}$) without necessarily terminating on every finite instance (over $\mathbf{S} \cup \mathbf{T}$).

**Critical Instance.** Let $\Sigma$ be a set of dependencies and let $*$ be a fresh constant ($* \notin \mathsf{Cst}(\Sigma)$). We define the critical instance $I_\Sigma$ of $\Sigma$ as the set of all the facts $R(t_1, \ldots, t_n)$ such that $R$ is a predicate symbol occurring in $\Sigma$ (i.e., $R \in \sigma_\Sigma$) and $\{t_1, \ldots, t_n\} \subseteq \mathsf{Cst}(\Sigma) \cup \{*\}$.

PROPOSITION 3.4 ([17]). *For every finite set of tgds $\Sigma$, the following statements are equivalent:*

- *$\Sigma$ ensures oblivious termination ($\Sigma \in \mathsf{TOC}$);*
- *$P_\Sigma$ terminates on the critical instance $I_\Sigma$;*
- *there exists a finite bound $k$ depending only on $\Sigma$ such that, for any instance $I$ over $\mathbf{S} \cup \mathbf{T}$ and any term $t$ occurring in $P_\Sigma(I)$, the size $|t|$ of $t$ is at most $k$.*

It follows from this proposition that, given a set of tgds and egds $\Sigma \in \mathsf{TOC}$ , and given a instance $I$, we can compute the universal model $P_\Sigma(I)$ in time $O(|\mathsf{Dom}(I)|^k)$ for some $k$ depending only on $\Sigma$. Combined with Proposition 3.2 we have the following theorem.

THEOREM 3.1. *For every strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ without disjunction, and more generally, for every finite set $\Sigma$ of tgds and egds, if $\Sigma \in \mathsf{TOC}$ the following problem is in* NP*: given a tgd $\delta$, does $\Sigma \models \delta$ ? Moreover, the problem becomes* PTIME *if $\delta$ is an egd or if $\delta$ is a tgd with a bounded number of atoms in its head.*

Since containment of CQ queries is NP-complete [1] and can be viewed as an implication problem, NP-hardness is also easily verified.

## 3.3 Supporting Disjunction

In this section, we extend our framework to also support disjunctive tgds and egds. In the case of disjunctive tgds (and egds) it as been shown in [4] that there is generally no (single) universal model and the more appropriate notion of *universal model set* was introduced in [4]. Given an instance $I$ and a set of disjunctive tgds $\Sigma$, a universal model set is a set of instances $\mathbf{J} = \{J_0, J_1, J_2, \ldots\}$ such that (i) for all $i \geq 0$ we have $J_i \models \Sigma$ and $I \xrightarrow{\text{hom}} J_i$, and (ii) for every instance $J'$ such that $J' \models \Sigma$ and $I \xrightarrow{\text{hom}} J'$ there exists some $i \geq 0$ such that $J_i \xrightarrow{\text{hom}} J'$. We denote by $\mathsf{UModS}(\Sigma, I)$ the universal model sets for $\Sigma$ and $I$.

We now generalize Proposition 3.2 to the case of disjunctive tgds and egds by considering universal models sets and extending the definitions of $s(\Sigma)$, $B_\delta$ and $H_\delta$ in a natural way. More precisely: the singularisation of a disjunctive tgd $\phi \rightarrow \bigvee_i \phi_i$ is simply defined as $\mathsf{sing}(\phi) \rightarrow \bigvee_i \psi_i$; $B_\delta$ is defined exactly as in the case of tgds and egds; and $H_\delta$ is the union of boolean conjunctive queries corresponding to the head of $\delta$ (while replacing "=" by "$E$" in the case of disjunctive egds).

PROPOSITION 3.5. *Given a finite set $\Sigma$ of disjunctive tgds and egds, given a disjunctive tgd or egd $\delta$, and given a universal model set $\mathbf{K} \in \mathsf{UModS}(s(\Sigma), B_\delta)$, we have $\Sigma \models \delta$ iff for every $K \in \mathbf{K}$ one of the following statements hold:*

- *$K^+ \models H_\delta$, or*
- *$\exists\, c, c' \in \mathsf{Cst}(\Sigma \cup \delta),\ c \neq c' \ \wedge\ E(c, c') \in K^+$.*

We next generalize the class $\mathsf{TOC}$, defined only for tgds and egds, to the case of disjunctive dependencies. For this, we first generalize the process of Skolemization. Given a set $\Sigma$ of disjunctive tgds we define the skolemization $P_\Sigma$ of $\Sigma$ as the set of rules of the form $B^s \rightarrow H_1^s \vee \cdots \vee H_k^s$ such that $(B \rightarrow H_1 \vee \cdots \vee H_k) \in \Sigma$ and each $(B^s \rightarrow H_i^s)$ corresponds to the standard skolemization of the tgd $(B \rightarrow H_i)$. Given a logic program $P_\Sigma$ as above and an instance $I$ we define a *$P_\Sigma$-derivation* of $I$ as a (possibly infinite) series $(I_0, I_1, \ldots, I_i, \ldots)$ such that $I_0 = I$ and for all $i > 0$ there exists a *justification* $(r_i, \bar{c}_i, j_i)$ of $I_i$ such that:

- $r_i$ is a rule $B^s(\overline{x}) \rightarrow H_1^s(\overline{x}) \vee \cdots \vee H_k^s(\overline{x})$ in $P_\Sigma$;
- $\bar{c}_i \in \mathsf{Dom}(I_{i-1})^{|\overline{x}|}$; and $B^s(\bar{c}_i) \subseteq I_{i-1}$;
- $j_i \in \{1, \ldots, k\}$ and $I_i = I_{i-1} \cup H_{j_i}^s(\bar{c}_i)$
- there is no $i' < i$ such that $(r_{i'}, \bar{c}_{i'}) = (r_{i'}, \bar{c}_{i'})$

Given a finite set $\Sigma$ of disjunctive tgds and an instance $I$ we say that $P_\Sigma$ *terminates on* $I$ iff all the $P_\Sigma$-derivations of $I$ are finite. We say that a finite $P_\Sigma$ derivation $(I_0, I_1, \ldots, I_n)$ of $I$ *leads to* $I_n$ iff there is no $I_{n+1} \supset I_n$ such that $(I_0, I_1, \ldots, I_n, I_{n+1})$ is a $P_\Sigma$-derivation of $I$. We finally denote by $P_\Sigma(I)$ the sets of instances $J$ such that some $P_\Sigma$-derivation of $I$ leads to $J$.

PROPOSITION 3.6. *For every finite set $\Sigma$ of disjunctive tgds and every instance $I$ such that $P_\Sigma$ terminates on $I$, we have $P_\Sigma(I) \in \mathsf{UModS}(\Sigma, I)$.*

DEFINITION 3.2. *We say that a finite set $\Sigma$ of disjunctive tgds and egds ensures oblivious termination, denoted by $\Sigma \in \mathsf{DTOC}$, iff $P_{s(\Sigma)}$ terminates on every finite instance $I$.*

PROPOSITION 3.7. *For any finite sets $\Sigma$ of disjunctive tgds, the following statements are equivalent:*

- *$\Sigma$ ensures oblivious termination ($\Sigma \in \mathsf{DTOC}$);*
- *$P_\Sigma$ terminates on the critical instance $I_\Sigma$;*
- *there exists a bound $k$, depending only on $\Sigma$ such that, for any instances $I$ and any term $t$ occurring in a $P_\Sigma$-derivation of $I$, the size $|t|$ of $t$ is at most $k$.*

From this, the generalization of Proposition 3.1 to disjunctive constraints easily follows:

THEOREM 3.2. *For every strong schema-mappings $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, and more generally for every set $\Sigma$ of disjunctive tgds and egds, if $\Sigma \in \mathsf{DTOC}$ the following problem is in $\Pi_2^p$: given a disjunctive tgd $\delta$, does $\Sigma \models \delta$? Moreover, the problem is in* coNP *if $\delta$ is a disjunctive egd or a disjunctive tgd where each head has a bounded number of atoms.*

Matching lower bounds can be obtained by a reduction from the containment of UCQ queries, which is $\Pi_2^p$-complete [20], and the tautology problem, which is coNP-complete.

## 3.4 Supporting Master Data

In this section we consider the problem of implication in the case where a source instance $I$ is immutable while the target instance $J$ can be any solution. In the context of strong schema-mappings, this setting corresponds to a situation of *semi-dynamic* data-exchange where the target instance $J$

can evolve in time after its creation while the initial source instance $I$ is not subject to evolution. This situation occurs, e.g., when $I$ is so-called *master data* [8] which is assume to be complete and consistent. As illustrated in the introduction, master data together with target-to-source constraints allow to provide an upper bound on the set of true facts in solutions. Note that in this section the distinction between the source schema and the target schema is crucial.

DEFINITION 3.3. *Given a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, a source instance $I$, and a constraint $\delta$ over $\mathbf{T}$, we write $\mathcal{M} \models^{\mathsf{sol}[I]} \delta$ iff for every target instance $J$ such that $(I, J) \models \Sigma$ (that is, every $J \in \mathsf{Sol}(\mathcal{M}, I)$) we have $J \models \delta$.*

We first illustrate on an example the additional source of difficulty brought by the introduction of (immutable) master data.

EXAMPLE 3.5. Consider the following strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ where $\Sigma \in \mathsf{TOC}$ and the following source instance $I$.

$$\mathbf{S} = \{R, S\} \text{ and } \mathbf{T} = \{A, B, C\}$$

$$\Sigma = \left\{ \begin{array}{l} R(x,y) \rightarrow A(x,y) \wedge B(y) \\ A(x,y) \wedge B(y) \rightarrow \exists u, v\ C(x,u) \wedge A(u,v) \\ A(x,y) \rightarrow S(y) \end{array} \right\}$$

$$I = \{R(0,0); S(0)\} \quad \leftarrow \text{(immutable) master data.}$$

The set of solutions $\mathsf{Sol}(\mathcal{M}, I)$ is characterized by the following set of tgds and egds

$$\Sigma' \equiv \left\{ \begin{array}{l} \rightarrow A(0,0) \wedge B(0) \\ A(x,y) \wedge B(y) \rightarrow \exists u, v\ C(x,u) \wedge A(u,v) \\ A(x,y) \rightarrow y = 0 \end{array} \right\}$$

$$\equiv \left\{ \begin{array}{l} \rightarrow A(0,0) \wedge B(0) \\ A(x,0) \rightarrow \exists u\ C(x,u) \wedge A(u,0) \\ A(u,v) \rightarrow v = 0 \end{array} \right\}$$

but the set of constraints $\Sigma'$ ensures no form of termination because the universal models of $\Sigma'$ are generally infinite. In particular, $\mathsf{UMod}(s(\Sigma'), \emptyset)$ contains no finite instance. □

Worse still, the following is readily verified:

PROPOSITION 3.8. *The following problem is undecidable: given a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, $\Sigma \in \mathsf{TOC}$, a source instance $I$ and a target tgd $\delta$, does $\mathcal{M} \models^{\mathsf{sol}[I]} \delta$ ?*

**Including the Source Instance.** Given a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ where $\Sigma = \Sigma_{st} \uplus \Sigma_{ts} \uplus \Sigma_s \uplus \Sigma_t$ and given a source instance $I$ for $\mathbf{S}$, we denote by $\Sigma[I]$ the set of disjunctive tgds and egds over $\mathbf{T}$ such that:

- $\Sigma[I]$ contains $\Sigma_t$
- $\Sigma[I]$ contains the set $\Sigma_{st}[I]$ of tgds with an empty body (corresponding to boolean conjunctive queries) of the form $\rightarrow \exists \overline{z}\ \psi(\overline{c}, \overline{z})$ such that there exists a tgd $\phi(\overline{x}, \overline{y}) \rightarrow \exists \overline{z}, \psi(\overline{x}, \overline{z})$ in $\Sigma_{st}$ and a tuple of constants $\overline{c} \in \mathsf{Dom}(I)^{|\overline{x}|}$ satisfying $I \models \exists \overline{y}, \phi(\overline{c}, \overline{y})$
- $\Sigma[I]$ contains the set $\Sigma_{ts}[I]$ of disjunctive egds

$$\begin{array}{rl} \phi(x_1, \ldots, x_n, \overline{y}) \rightarrow & (x_1 = c_1^1 \wedge \cdots \wedge x_n = c_n^1) \\ \vee & \cdots \vee (x_1 = c_1^p \wedge \cdots \wedge x_n = c_n^p) \end{array}$$

such that there exists a disjunctive tgd

$$\phi(x_1, \ldots, x_n, \overline{y}) \rightarrow \bigvee_{i=1}^{k} \left(\exists \overline{z}_i \psi_i(x_1, \ldots, x_n, \overline{z}_i)\right)$$

in $\Sigma_{ts}$ such that $\{(c_1^j, \ldots, c_n^j) | j \leq p\}$ corresponds to the set of tuples $(c_1, \ldots, c_n)$ satisfying

$$I \models \bigvee_{i=1}^{k} \left(\exists \overline{z}_i \psi_i(c_1, \ldots, c_n, \overline{z}_i)\right)$$

PROPOSITION 3.9. *For any strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, any source instance $I$ and any target constraint $\delta$ we have $\mathcal{M} \models^{\mathsf{sol}[I]} \delta$ iff $\Sigma[I] \models \delta$.*

It follows easily from Theorem 3.2 that $\mathcal{M} \models^{\mathsf{sol}[I]} \delta$ becomes decidable under the assumption that $\Sigma[I] \in \mathsf{DTOC}$. A very natural question however remains: how to ensure decidability (and reasonable data-complexity) when the source instance $I$ is not known in advance? The key contribution of this section is the following observation: it is possible to test the worst case scenario to ensure not only that $\Sigma[I] \in \mathsf{DTOC}$ for every source instance $I$, but also that $\Sigma[I]$ ensures fairly fast termination, even for a large source instance $I$. In the following definition we denote by $I_{\mathcal{M}}$ the *critical source instance* of $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ consisting of all the facts $R(t_1, \ldots, t_n)$ such that $R \in \mathbf{S}$ and $\{t_1, \ldots, t_n\} \subseteq \mathsf{Cst}(\Sigma) \cup \{*\}$.

DEFINITION 3.4. *Given a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ we write $\mathcal{M} \in \mathsf{MTOC}$ iff $\Sigma[I_{\mathcal{M}}] \in \mathsf{DTOC}$.*

PROPOSITION 3.10. *For any strong schema-mapping $\mathcal{M} \in \mathsf{MTOC}$ there exists a finite $k$, depending only on $\mathcal{M}$ such that, for every source instance $I$, every instance $J$, and every term $t$ occurring in a $P_{s(\Sigma[I])}$-derivation of $J$, the size $|t|$ of $t$ is bounded by $k$.*

THEOREM 3.3. *For any strong schema-mapping $\mathcal{M} \in \mathsf{MTOC}$ the following problem is in $\Pi_2^p$: given a source instance $I$ and a target disjunctive tgd $\delta$, does $\mathcal{M} \models^{\mathsf{sol}[I]} \delta$? The problem is in* coNP *if $\delta$ is a disjunctive egd or a disjunctive tgd with a bounded number of atoms in each head.*

Matching lower bounds can be obtained in a similar way as for Theorem 3.2.

# 4. DETERMINACY AND COMPLETENESS

The goal of this section is three-fold: First, we rephrase the notions of *relative information completeness* [8] and *determinacy* [21, 19] in the setting of strong schema-mappings; Second, we introduce interesting variations of these problems; and finally, we establish complexity results for all of these problems, hereby leveraging the results of Section 3.

## 4.1 Relative Information Completeness

In [8], the problem whether an instance $J$ has complete information to answer a query $Q$ was investigated. More precisely, $J$ is said to be complete for $Q$ if adding tuples to $J$

does not change the query result $Q(J)$. This problem becomes even more challenging when constraints are in place that limit which tuples can be added to $J$. In [8], these extensions of $J$ should adhere to certain containment constraints of the form $q(J) \subseteq \pi_X(I)$, where $q$ is a CQ query and $\pi_X(I)$ is a projection of a fixed (immutable) *master data*. As mentioned in the introduction, master data represents an instance which contains complete and consistent information. It is used in this context to provide an "upper bound" for the information extracted by $q(J)$. An instance $J$ is then said to be *complete for $Q$ relative to a set of containment constraints $\Sigma$ and master data $I$*, if for any $J' \supseteq J$ such that $J' \models \Sigma$, we have that $Q(J) = Q(J')$. In other words, adding tuples to $J$ either violates the constraints or does not change the query result $Q(J)$.

The following definition generalizes the notion of relative completeness, by incorporating **(i)** more general constraints from $J$ to $I$; and **(ii)** allowing constraints from $I$ to $J$; and **(iii)** accommodating for tgds and egds on the schema of $J$.

DEFINITION 4.1. *Given a strong schema-mapping $\mathcal{M}$, a source instance $I$, a solution $J \in \mathsf{Sol}(\mathcal{M}, I)$ and a target query $Q$ we say that $J$ is complete for $Q$ relative to $(\mathcal{M}, I)$ if for any other solution $J' \in \mathsf{Sol}(\mathcal{M}, I)$ such that $J \subseteq J'$, we have that $Q(J) = Q(J')$.*

It is easily verified (because of the target dependencies) that this problem is undecidable in general. We obtain decidability, however, when restricting $\mathcal{M}$ to the class $\mathsf{MTOC}$.

THEOREM 4.1. *Let $\mathcal{M}$ be a strong schema-mapping such that $\mathcal{M} \in \mathsf{MTOC}$, instances $I$ and $J$ and target query $Q$, deciding whether $J$ is complete for $Q$ relative to $(\mathcal{M}, I)$ is in $\Pi_2^p$. The problem is in coNP when $Q$ has a bounded number of atoms.*

This generalizes the $\Pi_2^p$-completeness result reported in [8]. Indeed, it is easily verified that the containment constraints studied there belong to $\mathsf{MTOC}$. A matching coNP lower bound follows from a reduction from the tautology problem.

We are also interested the following two stronger notions of completeness:

DEFINITION 4.2. *Given a strong schema-mapping $\mathcal{M}$, a source instance $I$ and a target query $Q$ we say that (1) $Q$ is complete relative to $(\mathcal{M}, I)$ if any $J \in \mathsf{Sol}(\mathcal{M}, I)$ is complete for $Q$ relative to $(\mathcal{M}, I)$; (2) $Q$ is universally complete relative to $\mathcal{M}$ iff $Q$ is complete relative to $(\mathcal{M}, I)$ for every source instance $I$.*

It is again easily verified that these two problems are undecidable in general. In order to obtain decidability, we proceed as follows: we first reduce the above two problems to an implication problem; this requires the modification of the input schema-mapping $\mathcal{M}$ into a new schema-mapping $\mathcal{M}^{2,\subseteq}$; the decidability then follows by assuming that $\mathcal{M}^{2,\subseteq}$ belongs to one of the classes introduced in Section 3.

Given a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ we define $\mathcal{M}^{2,\subseteq} = (\mathbf{S}, \mathbf{T}_1 \cup \mathbf{T}_2, \Sigma_1 \cup \Sigma_2 \cup \Sigma_\subseteq)$, where for $i = 1, 2$:

- $\mathbf{T}_i$ contains a predicate $R_i$ for every $R \in \mathbf{T}$;
- $\Sigma_i$ contains, for every $r \in \Sigma$, the constraint obtained from $r$ by replacing every atom $R(\bar{t})$ such that $R \in \mathbf{T}$ by the atom $R_i(\bar{t})$.
- $\Sigma_\subseteq$ contains a tgd $R_1(x_1, \ldots, x_n) \rightarrow R_2(x_1, \ldots, x_n)$ for every $R \in \mathbf{T}$.

Intuitively, $\mathbf{T}_1$ represents $J \in \mathsf{Sol}(\mathcal{M}, I)$ while $\mathbf{T}_2$ represents $J' \in \mathsf{Sol}(\mathcal{M}, I)$ such that $J' \supseteq J$. In view of the monotonicity of CQ queries, it suffices to check whether $Q(J') \subseteq Q(J)$. The key observation is that this containment test reduces to to checking whether a certain target constraint holds. From the results in Section 3, we then obtain:

THEOREM 4.2. *For every strong schema-mapping $\mathcal{M}$,*

1. *If $\mathcal{M}^{2,\subseteq} \in \mathsf{TOC}$ (no disjunction), then deciding whether $Q$ is universally complete relative to $\mathcal{M}$ is in* NP. *The problem is in* PTIME *if $Q$ has a bounded number of atoms.*
2. *If $\mathcal{M}^{2,\subseteq} \in \mathsf{DTOC}$, then deciding whether $Q$ is universally complete relative to $\mathcal{M}$ is in $\Pi_2^p$. The problem is in* coNP *if $Q$ has a bounded number of atoms.*
3. *If $\mathcal{M}^{2,\subseteq} \in \mathsf{MTOC}$, then deciding whether $Q$ is complete relative to $(\mathcal{M}, I)$ is in $\Pi_2^p$. The problem is in* coNP *if $Q$ has a number of atoms.*

## 4.2 Determinacy

A similar strategy can be followed for the determinacy problem [21, 19]. Recall that determinacy concerns the question whether a query $Q$ can be answered using a set $\mathbf{V}$ of views. More specifically, one says that $\mathbf{V}$ *determines* $Q$ iff $\mathbf{V}(J) = \mathbf{V}(J')$ implies $Q(J) = Q(J')$, for any two instances $J, J'$. In other words, determinacy says that $\mathbf{V}$ provides enough information to uniquely determine the answer to $Q$.

We propose the following generalization of determinacy:

DEFINITION 4.3. *Given a strong schema-mapping $\mathcal{M}$, a source instance $I$, and a target query $Q$ we say that (1) $Q$ is determined by $(\mathcal{M}, I)$ iff for all $J_1, J_2 \in \mathsf{Sol}(\mathcal{M}, I)$ we have $Q(J_1) = Q(J_2)$; and (2) $Q$ is determined by $\mathcal{M}$ iff $Q$ is determined in $(\mathcal{M}, I')$ for every source instance $I'$.*

When the views $\mathbf{V}$ are CQ queries, one can find an $\mathcal{M}$ such that $J \in \mathsf{Sol}(\mathcal{M}, I)$ iff $I = \mathbf{V}(J)$ (and similarly when $\mathbf{V}$ are UCQ queries). Hence, the above definition indeed generalizes the standard notion of determinacy.

It is readily verified that it is undecidable to check whether $Q$ is determined by $\mathcal{M}$ or $(\mathcal{M}, I)$. Similar to the completeness problem, we can see determinacy as an implication problem. Indeed, for a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ we let $\mathcal{M}^2$ be the strong schema-mapping $\mathcal{M}^2 = (\mathbf{S}, \mathbf{T}_1 \cup \mathbf{T}_2, \Sigma_1 \cup \Sigma_2)$ where for $i = 1, 2$, $\mathbf{T}_i$ and $\Sigma_i$ are defined as in $\mathcal{M}^{2,\subseteq}$. Again, $\mathbf{T}_1$ represents $J \in \mathsf{Sol}(\mathcal{M}, I)$ while $\mathbf{T}_2$ represents $J' \in \mathsf{Sol}(\mathcal{M}, I)$. In contrast to the completeness problem, no further constraints between $J$ and $J'$ are enforced. Checking determinacy now pours down to checking whether $Q(J) \subseteq Q(J')$ and $Q(J') \subseteq Q(J)$. That is, one needs to check the implication of two target constraints.

THEOREM 4.3. *For every strong schema-mapping $\mathcal{M}$,*

1. *If $\mathcal{M}^2 \in \mathsf{TOC}$ (no disjuntion), then deciding whether $Q$ is determined by $\mathcal{M}$ is in* NP. *The problem is in* PTIME *if $Q$ has a bounded number of atoms.*
2. *If $\mathcal{M}^2 \in \mathsf{DTOC}$, then deciding whether $Q$ is determined by $\mathcal{M}$ is in $\Pi_2^p$. The problem is in* coNP *if $Q$ has a bounded number of atoms.*
3. *If $\mathcal{M}^2 \in \mathsf{MTOC}$, then deciding whether $Q$ is determined by $(\mathcal{M}, I)$ is in $\Pi_2^p$. The problem is in* coNP *if $Q$ has a bounded number of atoms.*

## 4.3 Example

In this section, we illustrate determinacy and universal completeness. Consider the following strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st} \cup \Sigma_t \cup \Sigma_{ts} \cup \Sigma_s)$,

$$
\begin{aligned}
\mathbf{S} &= \{ \quad \mathsf{Phone}(\text{name,phone-nb}), \mathsf{Age}(\text{name,age}) \ \} \\
\mathbf{T} &= \{ \quad \mathsf{Employee}(\text{name,phone-nb,age}) \ \} \\
\Sigma_{st} &= \{ \quad \mathsf{Phone}(n,p) \rightarrow \exists a, \mathsf{Employee}(n,p,a) \\
&\qquad \mathsf{Age}(n,a) \rightarrow \exists p, \mathsf{Employee}(n,p,a) \ \} \\
\Sigma_t &= \{ \quad \mathsf{Employee}(n,p,a) \wedge \mathsf{Employee}(n,p',a') \rightarrow p = p' \\
&\qquad \mathsf{Employee}(n,p,a) \wedge \mathsf{Employee}(n,p',a') \rightarrow a = a' \} \\
\Sigma_{ts} &= \{ \quad \mathsf{Employee}(n,x,y) \rightarrow \\
&\qquad\qquad (\exists a, \mathsf{Age}(n,a)) \vee (\exists p, \mathsf{Phone}(n,p)) \ \} \\
\Sigma_s &= \{ \quad \mathsf{Age}(n,a) \wedge \mathsf{Age}(n,a') \rightarrow a = a' \\
&\qquad \mathsf{Phone}(n,p) \wedge \mathsf{Phone}(n,p') \rightarrow p = p' \ \}
\end{aligned}
$$

We can check that $\mathcal{M}^2$ and $\mathcal{M}^{2,\subseteq}$ both belong to $\mathsf{DTOC}$ (by evaluating the corresponding logical programs on the critical instance). Consequently, the algorithms behind Theorems 4.2 and 4.3 allow us to verify that

$$Q_1 = \{(x),\ \exists p, a, \mathsf{Employee}(x,p,a)\}$$

is determined by $\mathcal{M}$, while the query

$$Q_2 = \{(p),\ \exists n, a, \mathsf{Employee}(n,p,a)\}$$

is universally complete relative to $\mathcal{M}$ without being determined by $\mathcal{M}$.

To be more precise, $Q_1$ is determined in $\mathcal{M}$ because $Q_1$ is equivalent to $\{(x), (\exists a, \mathsf{Age}(x,a)) \vee (\exists p, \mathsf{Phone}(x,p))\}$. The query $Q_2$ is universally complete relative to $\mathcal{M}$ for the following reason: Let $J \in \mathsf{Sol}(\mathcal{M}, I)$ for some $I$. If we add an atom $\mathsf{Employee}(k,\ell,m)$ in $J$ for some new $\ell \notin Q_2(J)$, then by $\Sigma_{st}$ either $\mathsf{Age}(k,m)$ or $\mathsf{Phone}(k,\ell)$ must already exist in $I$. This in turn implies that $\mathsf{Employee}$ already contains a tuple $\mathsf{Employee}(k,\ell',m')$. However, $\Sigma_t$ enforces that $\ell = \ell'$ and $m = m'$. In other words, no new tuple can be added to $\mathsf{Employee}$. The reason why $Q_2$ is not determined is the following: for the solution $(\{\mathsf{Age}(n,a)\}, \{\mathsf{Employee}(n, \#_0, a)\})$ where $n, a \in \mathsf{Cst}$ and $\#_0 \in \mathsf{Nulls}$ the value of $\#_0$ is not determined. For instance $(\{\mathsf{Age}(n,a)\}, \{\mathsf{Employee}(n, \#_1, a)\})$ is another solution. □

# 5. TARGET-TO-SOURCE CONSTRAINTS

So far, we have considered the implication problem in the context of *strong* schema-mappings which (possibly) contain target-to-source dependencies. As discussed in the introduction, enriching standard schema-mappings with target-to-source dependencies requires some care and raises several questions.

In particular, in the context of dynamic data-exchange, and when assuming that a set $\Sigma_{ts}$ of some target-to-source dependencies are part of the input, it is indeed important to make sure that these target-to-source dependencies, meant to constrain the evolution of the target instances, are satisfied at the initial stage of the exchange, when a first target instance $J$ is computed from the initial source instance $I$ and with a standard schema-mapping management system (SMMS).

More formally, given a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ where [1] $\Sigma = \Sigma_{st} \cup \Sigma_t \cup \Sigma_{ts}$ and given two instances $I$ and $J$ such that

- $I$ is a source instance of $\mathbf{S}$;
- $J$ is a universal solution in $\mathsf{USol}(\mathcal{M}, I)$ computed by some SMMS using only the standard schema-mapping $\mathcal{M}' = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$, without taking $\Sigma_{ts}$ into account,

we want to make sure that $(I, J) \models \Sigma_{ts}$ will be true, before knowing the source instance $I$, and without necessarily knowing precisely which SMMS is to be used. In other words, we want to decide whether $\Sigma_{ts}$ is "implied" by the (standard) schema-mapping embedded in the strong schema-mapping.

Note that here we ignore the source dependencies $\Sigma_s$. Indeed, it is assumed that every given source instance $I$ already satisfies $\Sigma_s$.

As already mentioned in the introduction, the results of Section 3 are of no use in this case since no target-to-source constraints can be logically implied by a standard schema-mapping. We therefore set out to define alternative semantics of entailment with decidable properties.

## 5.1 Universal Solutions and Cores

In this section, we first consider the two most popular notions of semantics for schema-mappings based $\mathsf{USol}(\mathcal{M}, I)$ (the set of *universal solutions*) and $\mathsf{Core}(\mathcal{M}, I)$ (the set of *cores of universal solutions*). We refer to Section 2 for the definitions of these notions.

DEFINITION 5.1. *Given a schema-mapping $\mathcal{M}$ and a target-to-source dependency $\delta$ we write*

- $\mathcal{M} \models^{\mathbf{usol}} \delta$ *iff for every source instance $I$ and every $J \in \mathsf{USol}(\mathcal{M}, I)$ we have $(I, J) \models \delta$, and*
- $\mathcal{M} \models^{\mathbf{core}} \delta$ *iff for every source instance $I$ and every $J \in \mathsf{Core}(\mathcal{M}, I)$ we have $(I, J) \models \delta$.*

In the light of the following example (similar to examples used in [16, 13]), is becomes clear that the notion of universal solution is not restrictive enough because **(i)** some universal solutions are neither natural nor realistic; and **(ii)** USol leads to a notion of entailment $\models^{\mathbf{usol}}$ which is not general enough for target-to-source dependencies (intuitively, $\mathcal{M} \models^{\mathbf{usol}} \delta$ only holds for a trivial schema-mappings $\mathcal{M}$ or a trivial target-to-source $\delta$).

[1] Note that for the sake of simplicity (and tractability) we ignore here the set of source dependencies $\Sigma_s$ that may come with the source schema.

EXAMPLE 5.1. Consider $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ with $\mathbf{S} = \{A\}$, $\mathbf{T} = \{B\}$, $\Sigma_t = \emptyset$ and where $\Sigma_{st} = \{A(x) \rightarrow B(x)\}$ copies $A$ to $B$. Let $\delta$ be the source-to-target dependency $B(x) \rightarrow A(x)$. The "natural" solution for $\mathcal{M}$ and $I_n = \{A(1), \ldots, A(n)\}$ would be the target instance $J_n = \{B(1), \ldots, B(n)\}$ which satisfies $(I_n, J_n) \models \delta$.
We can observe, however, that the target instance $J'_n = \{B(1), \ldots, B(n), B(\#_0)\}$ where $\#_0 \in \mathsf{Nulls}$ is also a universal solution. Since $(I_n, J'_n) \not\models \delta$, we have $\mathcal{M} \not\models^{\mathbf{usol}} \delta$. In contrast, $\mathsf{Core}(\mathcal{M}, I_n)$ captures precisely the desired semantics since $\mathsf{Core}(\mathcal{M}, I_n) = \{J_n\}$ and we have $\mathcal{M} \models^{\mathbf{core}} \delta$. $\square$

It can easily be argued that $\models^{\mathbf{core}}$ is the most natural semantics for schema-mappings and also the most relevant semantics in the context of target-to-source dependencies. However, the following proposition shows that general and unrestricted use of $\models^{\mathbf{core}}$ easily leads to undecidability. This is even the case for schema-mappings that ensures oblivious termination ($\Sigma_{st} \cup \Sigma_t \in \mathsf{TOC}$), are without egds, without disjunction in $\delta$, and under the assumption that $\Sigma_t$ satisfies the syntactic criteria of *weak acyclicity* mentioned in the introduction and further discussed in Section 6.

PROPOSITION 5.1. *The following problem is undecidable: given a schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ where $\Sigma_t$ consists only of a weakly acyclic set of target tgds and given a target-to-source tgd $\delta$, does $\mathcal{M} \models^{\mathbf{core}} \delta$?*

## 5.2 Core-Ground and Core-Safe Queries

In this section, we identify more precisely the origin of the undecidability stated in Proposition 5.1. In particular, we show that decidability can be obtained by requiring $\delta$ to satisfy either a very general and high-level property ($\delta$ is *core-ground*) or a less general but tractable property ($\delta$ is *core-safe*).

**Core-ground queries.** Given a schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ and a conjunctive query $Q$ over $\mathbf{T}$ we say $Q$ is core-ground (in $\mathcal{M}$) iff for every source instance $I$ and every $J \in \mathsf{Core}(\mathcal{M}, I)$ there is no null in $Q(J)$ (we say that $Q(J)$ is ground). Given a target-to-source disjunctive tgd $\delta$ of the form $\phi(\overline{x}, \overline{y}) \rightarrow \bigvee_i H_i$, we then say that $\delta$ is core-ground iff the conjunctive query $Q_\phi = \{\overline{x} \mid \exists \overline{y}, \phi(\overline{x}, \overline{y})\}$ is core-ground.

PROPOSITION 5.2. *The following problem is undecidable: given a schema-mapping $\mathcal{M} \in \mathsf{TOC}$ and a target query of the form $Q = \{(\overline{x}) | R(x)\}$ for some unary predicate $R$, is $Q$ core-ground in $\mathcal{M}$ ?*

THEOREM 5.1. *The following problem is decidable: given a schema-mapping $\mathcal{M} \in \mathsf{TOC}$ and a given a target-to-source disjunctive tgds $\delta$ such that $\delta$ is core-ground in $\mathcal{M}$, does $\mathcal{M} \models^{\mathbf{core}} \delta$ ?*

We next identify a decidable syntactic criterion ensuring that a target query is core-ground in a schema-mapping.

**Core-safe Queries.** Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ be a schema-mapping. We define a *position* as a pair $(R, i)$ preferably writen $R^i$ where $R \in \mathbf{T}$ and $i \in \{1, \ldots, a_R\}$. Given a set of atoms $\phi$ over $\mathbf{T}$ and a variable $x$ we denote by $\mathcal{P}_{\phi,x}$ the set of positions $R^i$ such that $\phi$ contains an atom $R(t_1, \ldots, t_n)$ where $t_i = x$. We say that a position $p$ of $\mathbf{T}$ is *initially affected* in $\mathcal{M}$ iff there is a tgd $\phi(\overline{x}, \overline{y}) \rightarrow \exists \overline{z},\ \psi(\overline{x}, \overline{z})$ in $\Sigma_{st} \cup \Sigma_t$ such that $p \in \cup_{z \in \overline{z}} \mathcal{P}_{\psi,z}$. We let $\mathsf{IA}(\mathcal{M}) = \{\{p_1\}, \ldots, \{p_m\}\}$ where $\{p_1, \ldots, p_n\}$ is the set of initially affected positions.

Let $P = \{P_1, P_2, \ldots, P_m\}$ be a finite set where each $P_i$ is a set of positions. Given a target tgd $\phi(\overline{x}, \overline{y}) \rightarrow \exists \overline{z},\ \psi(\overline{x}, \overline{z})$ in $\Sigma$ and given a universal quantified variable $x \in \overline{x}$ such that $\mathcal{P}_{\phi,x} \subseteq P_1$ we write $P \rightsquigarrow^{\mathbf{tgd}}_{\mathcal{M}} \{P_1 \cup \mathcal{P}_{\psi_x}, P_2, \ldots, P_n\}$. Given a target egd of the form $\phi(x_1, x_2, \overline{y}) \rightarrow x_1 = x_2$ in $\Sigma$ such that $\mathcal{P}_{\phi,x_1} \subseteq P_1$ and $\mathcal{P}_{\phi,x_2} \subseteq P_2$ we write $P \rightsquigarrow^{\mathbf{egd}}_{\mathcal{M}} \{P_1 \cup P_2, P_3, \ldots, P_m\}$. We finally denote by $\rightsquigarrow_{\mathcal{M}}$ the reflexive and transitive closure of the binary relation $\rightsquigarrow^{\mathbf{tgd}}_{\mathcal{M}} \cup \rightsquigarrow^{\mathbf{egd}}_{\mathcal{M}}$.

Given a conjunctive query $Q = \{\overline{x} | \exists \overline{y}, \phi(\overline{x}, \overline{y})\}$ over $\mathbf{T}$, we say that $Q$ is unsafe in $\mathcal{M}$ iff there exists an $x \in \overline{x}$ and a set $\{P_1, \ldots, P_m\}$ such that (i) $\mathsf{IA}(\mathcal{M}) \rightsquigarrow^*_{\mathcal{M}} \{P_1, \ldots, P_m\}$; and (ii) $\mathcal{P}_{\phi,x} \subseteq P_1$. We say that $Q$ is *core-safe* in $\mathcal{M}$ iff $Q$ is not unsafe in $\mathcal{M}$. We say that a disjunctive tgd $\phi(\overline{x}, \overline{y}) \rightarrow \bigvee_i \exists \overline{z}_i,\ \psi_i(\overline{x}, \overline{z})$ is core-safe in $\mathcal{M}$ iff $Q_\phi$ is core-safe in $\mathcal{M}$.

EXAMPLE 5.2. Consider the two schema mappings $\mathcal{M}_a = (\mathbf{S}, \mathbf{T}, \{r_1; r_2; r_3\})$ and $\mathcal{M}_b = (\mathbf{S}, \mathbf{T}, \{r_1; r_2; r_3; e_1\})$, with $\mathbf{S} = \{A; B\}$, $\mathbf{T} = \{R; S\}$ and constraints:

$$r_1 = A(x) \rightarrow \exists y, z,\ R(x, y, z)$$
$$r_2 = B(x, y, z) \rightarrow R(x, y, z)$$
$$r_3 = R(x, y, y) \rightarrow S(x, y)$$
$$e_1 = R(x, y, z) \rightarrow y = z$$

Let $Q = \{x, y | S(x, y)\}$. Then, $\mathsf{IA}(\mathcal{M}_a) = \{\{R^1\}; \{R^2\}\}$ which is a fixed point for $\rightsquigarrow_{\mathcal{M}}$ and therefore $Q$ is core-safe in $\mathcal{M}_a$. In constrast, $Q$ is not core-safe in $\mathcal{M}_b$. Indeed,

$$\mathsf{IA}(\mathcal{M}_b) = \{\{R^1\}; \{R^2\}\} \rightsquigarrow^{\mathbf{egd}}_{\mathcal{M}} \{\{R^1; R^2\}\} \rightsquigarrow^{\mathbf{tgd}}_{\mathcal{M}} \{\{R^1; R^2; S^2\}\}$$

but $\mathcal{P}_{Q,y} = \{S^2\}$ is included in $\{R^1; R^2; S^2\}$. $\square$

PROPOSITION 5.3.

- *For all schema-mappings $\mathcal{M}$ and all target-to-source disjunctive tgds $\delta$, if $\delta$ is core-safe in $\mathcal{M}$, then $\delta$ is core-ground in $\mathcal{M}$ (and Theorem 5.1 is applicable).*
- *Moreover, the following problem is in PTIME: given $\mathcal{M}$ and $\delta$ as above, is $\delta$ core-safe in $\mathcal{M}$ ?*

## 5.3 Closed-World Assumption

In this section, we propose another semantics which is more flexible than $\models^{\mathbf{core}}$ and captures an intuitive notion of *closed-world semantics*, in a style similar to that of [15, 13]. Furthermore, under this semantics we obtain decidability for arbitrary target-to-source dependencies (in contrast to the safety assumption in the previous section).

**Canonical Instance.** Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma_{st}, \Sigma_t)$ be a schema-mapping. In the case of tgds only (without target egds) and given a source instance $I$ we define the *canonical solution* of $I$ as the target instance $\Pi_{\mathbf{T}}(P_\Sigma(I))$ where $P_\Sigma$ is the fixed point of $I$ by the Skolemisation $P_\Sigma$ of $I$ and $\Pi_{\mathbf{T}}$ is the operation selecting the atoms $R(\overline{t})$ such that $R \in \mathbf{T}$. From Proposition 3.3 we know that the canonical solution is indeed a solution (i.e., it belongs to $\mathsf{USol}(\mathcal{M}, I)$).

In the presence of egds, we use the substitution-free simulation defined in Section 3.1 to define a *canonical instance* $\mathcal{M}^{\mathsf{can}}(I)$ of $I$. That is, we define $\mathcal{M}^{\mathsf{can}}(I) = \Pi_{\mathbf{T}}(P_{s(\Sigma)}(I)^{+})$ where: $s(\Sigma)$ is the substitution-free simulation of $\Sigma$; $^{+}$ is the operation that applies the substitution axioms; and $\Pi_{\mathbf{T}}$ selects the facts that are over the target schema (dropping in particular the facts $E(t,t')$). It can be seen that $\mathcal{M}^{\mathsf{can}}(I)$ coincides precisely with the canonical solution of $I$ when there is no egd in $\Sigma$. We use, however, the expression canonical *instance* because in the presence of egds, $\mathcal{M}^{\mathsf{can}}(I)$ is not necessarily a solution. This is illustrated in the following example.

EXAMPLE 5.3. Consider the schema-mapping $\mathcal{M}$ where $\mathbf{S} = \{A, B\}$, $\mathbf{T} = \{R, S, T\}$ and $(\Sigma_{st}, \Sigma_t)$ is as follows.

$$\Sigma_{st} = \left\{ \begin{array}{l} A(x) \rightarrow \exists u, v\ R(x,u) \wedge R(v,x) \wedge S(u,v) \\ B(x) \rightarrow \exists R(x,x) \end{array} \right\}$$

$$\Sigma_t = \left\{ \begin{array}{l} S(x,y) \rightarrow x = y \\ R(x,x) \rightarrow T(x) \end{array} \right\}$$

In this setting, assuming that $P_{s(\Sigma)}$ is equal to

$$\left| \begin{array}{l} A(x) \rightarrow R(x, f_u\langle x\rangle) \wedge R(f_v\langle x\rangle, x) \wedge S(f_u\langle x\rangle, f_v\langle x\rangle) \\ B(x) \rightarrow R(x,x) \\ S(x,y) \rightarrow E(x,y) \\ R(x,x') \wedge E(x,y') \rightarrow T(x) \\ (+\text{axioms of } E) \end{array} \right.$$

we observe that for $I = \{A(0), B(1)\}$ the canonical instance

$$\begin{array}{rl} \mathcal{M}^{\mathsf{can}}(I) = \{ & R(0, f_u\langle 0\rangle); R(f_v\langle 0\rangle, 0); S(f_u\langle 0\rangle, f_v\langle 0\rangle); \\ & R(0, f_v\langle 0\rangle); R(f_u\langle 0\rangle, 0); S(f_v\langle 0\rangle, f_u\langle 0\rangle); \\ & R(1,1); T(1)\ \} \end{array}$$

violates $\Sigma_t$ and is not a solution. □

From Theorem 3.4 in [15], it follows that the definition below coincides with the notion of CWA-solutions introduced in [15] when only source-to-target tgds are present.

DEFINITION 5.2. *Given a schema-mapping $\mathcal{M}$ and a source instance $I$, we say that a universal solution $J \in \mathsf{USol}(\mathcal{M}, I)$ is CwA iff there exists a homomorphism $h$ from $\mathcal{M}^{\mathsf{can}}(I)$ to $J$ such that $J = h(\mathcal{M}^{\mathsf{can}}(I))$. We denote by $\mathsf{CwA}(\mathcal{M}, I)$ the set of CwA universal solutions in $\mathsf{USol}(\mathcal{M}, I)$ and, given a dependency $\delta$ we write $\mathcal{M} \models^{\mathbf{cwa}} \delta$ iff for every source instance $I$ and every $J \in \mathsf{CwA}(\mathcal{M}, I)$ we have $(I, J) \models^{\mathbf{cwa}} \delta$.*

We can first observe that $\models^{\mathbf{cwa}}$ is a good candidate for decidability in the sense that it is more restrictive than $\models^{\mathbf{usol}}$ while being less restrictive than $\models^{\mathbf{core}}$.

PROPOSITION 5.4. $\models^{\mathbf{usol}} \subsetneq \models^{\mathbf{cwa}} \subsetneq \models^{\mathbf{core}}$, *that is, for all schema-mappings $\mathcal{M}$ and all source instances $I$ we have*

$$\mathsf{Core}(\mathcal{M}, I) \subseteq \mathsf{CwA}(\mathcal{M}, I) \subseteq \mathsf{USol}(\mathcal{M}, I)$$

*and both inclusions can be strict.*

We are now ready to prove our final theorem.

THEOREM 5.2. *The following problem is decidable: given a schema-mapping $\mathcal{M} \in \mathsf{TOC}$ and given a target-to-source disjunctive tgd $\delta$, does $\mathcal{M} \models^{\mathbf{cwa}} \delta$ ?*

The following example shows that Theorem 5.2 can be used in scenarios that are not covered by the previous section.

EXAMPLE 5.4. Consider the schema-mapping $\mathcal{M}$ from Example 5.3 and consider the target-to-source tgd

$$\delta: \quad T(x) \rightarrow B(x)$$

It is clear that $\mathcal{M} \in \mathsf{TOC}$ but it can be seen that the query $\{x | T(x)\}$ is not core-safe (and Theorem 5.3 does not apply). We can nonetheless use the algorithm corresponding to Theorem 5.2 to check that we have $\mathcal{M} \models^{\mathbf{cwa}} \delta$ (and therefore $\mathcal{M} \models^{\mathbf{core}} \delta$). □

# 6. RELATED WORK

**Finite Models.** The different semantics considered in this paper rely on *infinite models*. In particular, for a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, we write $\Sigma \models \delta$ whenever, for every (possibly infinite) instances $I$ and $J$ over the source schema and target schema, respectively, we have $(I, J) \models \Sigma$ implies $(I, J) \models \delta$. As shown in [4], in the case of strong schema-mappings this notion of entailment is not equivalent to the one where only *finite* instances are considered. The reason why we did not discuss this distinction in the previous sections is the following observation (similar to Proposition 5 in [17]): the assumptions of termination considered in the Theorems of this paper all ensure both tractability and *finite controllability* (meaning that finite-model semantics and infinite-model semantics coincide). For instance, our notion of determinacy has been shown in [19] to differ from *finite* determinacy (for every *finite* solutions $J_1, J_2 \in \mathsf{Sol}(\mathcal{M}, I)$ we have $Q(J_1) = Q(J_2)$). However, if for a given $\mathcal{M}$, $\mathcal{M}^2 \in \mathsf{DTOC}$, then a target query $Q$ is determined by $\mathcal{M}$ iff it is finitely determined by $\mathcal{M}$.

**Weak Acyclicity.** It has been shown in [17] that the class TOC is a strict generalisation of the widely-used notion of *weak acyclicity* [6, 5], even in the presence of arbitrary egds. It can be seen that MTOC is similarly a strict generalization of *weak acyclicity*. More precisely, if we consider a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ where $\Sigma$ consists of tgds and egds only (no disjunction), then we can check that the following holds: if $\Sigma_t$ is the union of a weakly-acyclic set of tgds and an arbitrary set of egds, then $\mathcal{M} \in \mathsf{MTOC}$. Note that MTOC is at the same time a different kind of criterion than *weak-acyclicity*. Indeed, while *weak-acyclicity* of schema-mappings can be decided in PTIME, membership in MTOC is in general undecidable. Recall, however, that MTOC (just like TOC and DTOC) can be tested on a critical instance which is typically of small size and provided that "efficient" schema-mappings are given, the MTOC class appears feasible in practice. Finally, tractable subclasses of MTOC (and DTOC) can be defined in a similar way as in [17].

**Peer Data Exchange.** Target-to-source dependencies have been considered in [9]. There, a specific class of strong schema-mappings called Peer-Data-Exchange settings (PDE) was introduced. In particular, a PDE setting is a strong schema-mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ with no source constraints and no disjunction (where $\Sigma = \Sigma_{st} \cup \Sigma_{ts} \cup \Sigma_t$). Static analysis of the PDE setting was not the focus of [9]. Instead,

a very interesting decision problem, denoted SOL($\mathcal{M}$) was considered: given a source instance $I$ and a target instance $J$ for a fixed PDE setting $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, is there a target instance $J^+$ (called a *solution* in [9]) such that $J \subseteq J^+$ and $(I, J^+) \models \Sigma$? This problem SOL($\mathcal{M}$) has been proven in [9] to be NP-complete under the assumption of weak acyclicity, while it is clearly undecidable in general. Our results in Section 3 can be used to strengthen this result: SOL($\mathcal{M}$) is in NP whenever $\mathcal{M} \in \mathsf{MTOC}$. Note that other results (such as Theorem 2 in [9]) obtained under the assumption of weak acyclicity can similarly be generalized to the class MTOC.

**Restricted Chase**. The assumptions of termination in this paper rely on the oblivious chase, which differs from the well-known procedure called *the restricted chase* (often simply referred to as *the chase*). With the restricted chase, an inconsistency is resolved (intuitively, by adding the head of a tgd) only when necessary. We can observe that the restricted chase terminates whenever the oblivious chase terminates (while the converse does not hold). It can be seen however that the restricted chase (and similarly, the *core chase* introduced in [4]) is problematic for several reasons: (i) termination cannot be tested on the critical source instance, (ii) applying a rule is more costly than applying a rule of the oblivious chase, and (iii) many questions remain open (e.g. does termination of the restricted chase implies *polynomial* termination?). Note however that several decidable criteria such as the very involved *inductive restriction* in [18] (which is incomparable with TOC) have been designed to ensure termination of (only) the restricted chase. We intend to show in the future that our results can be extended to this class by using a rewriting algorithm, that given an inductively restricted set $\Sigma$ of tgds and egds, returns a set of tgds and egds $\Sigma' \in \mathsf{TOC}$.

**Guarded Tgds.** Another setting in which entailment of tgds and egds has been shown decidable (with a good complexity) is when all the tgds are *guarded* (and the egds are *separable*) [2]. A tgd is called guarded iff there is an atom in the body covering all the universal variables. For instance, the following tgd $\delta_1$ is guarded (by the atom $A(x, y)$) while the tgd $\delta_2$ is not guarded.

$$\delta_1 : \quad A(x, y) \wedge C(y) \rightarrow R(x, y)$$
$$\delta_2 : \quad A(x, y) \wedge B(y, z) \rightarrow R(x, z)$$

The main advantage of this setting is that it neither requires the existence of a *finite* universal model (in contrast with [4]), nor any assumption of termination. It is also interesting to observe that the target tgds used in real-life schema-mappings usually consist of inclusion dependencies, that is, of tgds which have only one atom in the body and are therefore guarded. One limit of this framework, however, is the fact that it does not apply to strong schema-mappings with non-guarded source-to-target tgds and target-to-source tgds. To be more precise, as soon as $\Sigma_{st}$ contains a single non-guarded tgd such as $\delta_2$ above (used to materialize the join of $A$ and $B$), the results of [2] cannot be applied directly.

**Hypertree-width**. We have shown that several complexity results can be significantly improved when assuming a fixed bound on the number of atoms, either in the head $H$ of the constraints $\delta$ considered in Section 3 (Theorems 3.1, 3.2 and 3.3), or in the conjunctive queries $Q$ considered in Section 4 (Theorems 4.1, 4.2 and 4.3). It can been seen that the same complexity results hold under the much more general assumption of *bounded hypertree-width* [10] which strictly covers by itself the cases where $H$ (or $Q$) has a bounded number of existential variables, is *acyclic* or has a bounded *tree-width* [10].

# 7. CONCLUSION

We have studied the implication problem for strong schema-mappings which are extensions of standard schema-mappings with (disjunctive) target-to-source and source dependencies. As an application, we revisited the problems of relative information completeness and determinacy, both instances of implication problems. We also addressed the question whether target-to-source constraints are already implied by the embedded standard schema-mapping, a problem that cannot be solved using the standard logical implication techniques mentioned above.

Interesting open problems concern the precise complexity of the implication problem of target-to-source constraints under both the core- and CwA-implication semantics. Other questions involve the identification of tractable subclasses of TOC, DTOC and MTOC (similar to those in [17]) and the generalisation to classes of dependencies ensuring (only) the termination of the restricted chase.

*Acknowledgment.* The first author was funded by EPSRC EP/E010865/1 and partially supported by ERC Advanced Grant Webdam on Foundation of Web data management. The second author was supported by EPSRC EP/E029213/1. We thank Serge Abiteboul and Alin Deutsch for helpful discussions.

# 8. REFERENCES

[1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. 1995.

[2] A. Calì, G. Gottlob, and T. Lukasiewicz. A general datalog-based framework for tractable query answering over ontologies. In *PODS*, pages 77–86, 2009.

[3] S. Chaudhuri and M. Y. Vardi. On the equivalence of recursive and nonrecursive datalog programs. *J. Comput. Syst. Sci.*, 54(1):61–78, 1997.

[4] A. Deutsch, A. Nash, and J. B. Remmel. The chase revisited. In *PODS*, pages 149–158, 2008.

[5] A. Deutsch and V. Tannen. Xml queries and constraints, containment and reformulation. *Theor. Comput. Sci.*, 336(1):57–87, 2005.

[6] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

[7] R. Fagin, P. G. Kolaitis, and L. Popa. Data exchange: getting to the core. *ACM Trans. Database Syst.*, 30(1):174–210, 2005.

[8] W. Fan and F. Geerts. Relative information completeness. In *PODS*, pages 97–106, 2009.

[9] A. Fuxman, P. G. Kolaitis, R. J. Miller, and W. C. Tan. Peer data exchange. *ACM Trans. Database Syst.*, 31(4):1454–1498, 2006.

[10] G. Gottlob, N. Leone, and F. Scarcello. Hypertree decompositions and tractable queries. *J. Comput. Syst. Sci.*, 64(3):579–627, 2002.

[11] G. Gottlob and A. Nash. Efficient core computation in data exchange. *J. ACM*, 55(2), 2008.

[12] L. M. Haas, M. A. Hernández, H. Ho, L. Popa, and M. Roth. Clio grows up: from research prototype to industrial tool. In *SIGMOD*, pages 805–810, 2005.

[13] A. Hernich and N. Schweikardt. Cwa-solutions for data exchange settings with target dependencies. In *PODS*, pages 113–122, 2007.

[14] P. G. Kolaitis. Schema mappings, data exchange, and metadata management. In *PODS*, pages 61–75, 2005.

[15] L. Libkin. Data exchange and incomplete information. In *PODS*, pages 60–69, 2006.

[16] L. Libkin and C. Sirangelo. Data exchange and schema mappings in open and closed worlds. In *PODS*, pages 139–148, 2008.

[17] B. Marnette. Generalized schema-mappings: from termination to tractability. In *PODS*, pages 13–22, 2009.

[18] M. Meier, M. Schmidt, and G. Lausen. On chase termination beyond stratification. *PVLDB*, 2(1):970–981, 2009.

[19] A. Nash, L. Segoufin, and V. Vianu. Determinacy and rewriting of conjunctive queries using views: A progress report. In *ICDT*, pages 59–73, 2007.

[20] Y. Sagiv and M. Yannakakis. Equivalences among relational expressions with the union and difference operators. *J. ACM*, 27(4):633–655, 1980.

[21] L. Segoufin and V. Vianu. Views and queries: determinacy and rewriting. In *PODS*, pages 49–60, 2005.

[22] O. Shmueli. Equivalence of datalog queries is undecidable. *J. Log. Program.*, 15(3):231–241, 1993.

[23] B. ten Cate, L. Chiticariu, P. G. Kolaitis, and W. C. Tan. Laconic schema mappings: Computing the core with sql queries. *PVLDB*, 2(1):1006–1017, 2009.