Union and Intersection of Schema Mappings

Jorge Pérez[1], Reinhard Pichler[2], Emanuel Sallinger[2], and Vadim Savenkov[2]

[1] Department of Computer Science, Universidad de Chile
[2] Faculty of Informatics, Vienna University of Technology

Abstract. Schema mappings have been extensively studied in database research over the past decade – notably in the areas of data exchange and data integration. Recently, the notion of an information transfer order on schema mappings has been introduced to compare the amount of source information that is actually transferred by two mappings. In this paper, we present two new operators: the *union* and *intersection* of mappings. The union of two mappings allows us to describe the sum of all information transferred by several mappings. The intersection refers to the common part of information transferred by several mappings. As one of our main results we prove that there exists a large class of mappings (containing the class of source-to-target tuple-generating dependencies) that forms a complete lattice with respect to these two operators.

1 Introduction

Schema mappings allow us to describe the relationship between two schemas. As such, schema mappings have been extensively studied in Data Exchange [9] and Data Integration [12]. Bernstein and Melnik [5, 6, 15] have proposed several fundamental operators on schema mappings, with composition [14, 7, 16, 2] and inverse [8, 10, 4, 3] being the most prominent ones. Recently, new concepts have been introduced [1, 10] to compare schema mappings in terms of the amount of source information transferred by the mappings. In this work, we present two new operators on mappings, which we consider as fundamental for studying the information transferred by several mappings. We thus introduce the *union* and *intersection* of mappings. The union of two mappings allows us to describe the sum of all information transferred by the mappings. The intersection of two mappings refers to the common part of information transferred by the mappings.

Before providing more details on our new schema mapping operators, we recall the notion of *information transfer* introduced by Arenas et al. [1]. Intuitively, the authors define a criterion to compare the amount of information that two mappings transfer from source to target. As an example consider the following two mappings given by source-to-target tuple-generating dependencies (st-tgds) between the source schema with one ternary relation A and the target schema with binary relations S and T and a ternary relation R:

$$\mathcal{M}_1 : A(x, y, z) \rightarrow \exists u\, S(x, u)$$
$$\mathcal{M}_2 : A(x, y, z) \rightarrow T(x, y)$$

Intuitively, $\mathcal{M}_2$ *transfers more information* than $\mathcal{M}_1$ since the first and the second component of tuples in A are being transferred to the target under $\mathcal{M}_2$, while only the first component is being transferred under $\mathcal{M}_1$ [1]. This intuition was formalized in [1] and several algorithmic properties were studied. In particular the authors show that given two mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ specified by st-tgds, it can be decided whether $\mathcal{M}_2$ transfers more information than $\mathcal{M}_1$ [1]. A similar notion of information transfer was proposed by Fagin et al. [10], and it has been shown that both notions coincide for the important case of mappings specified by st-tgds [1, 10].

A possible application of the information transfer notion is in automatic mapping-generation tools [11]. As described in [10], if two possible mappings are automatically generated by different tools, then a plausible criterion to decide which mapping is the better to be used, is to choose the mapping that transfers more information from source to target. But what happens if both tools generate *incomparable mappings* in terms of information transfer? Then the criterion presented in [1, 10] can no longer be used to decide which mapping to choose. This is one of the questions that motivate our research.

To illustrate our new schema mapping operators of *intersection* and *union*, consider the following mapping between the same schemas as $\mathcal{M}_1$ and $\mathcal{M}_2$:

$$\mathcal{M}_3 : A(x, y, z) \rightarrow S(x, z)$$

It can be shown that $\mathcal{M}_2$ and $\mathcal{M}_3$ are incomparable with respect to the information that both mappings transfer from source to target. Assume now that $\mathcal{M}_2$ and $\mathcal{M}_3$ are mappings that have been generated independently by two different tools. Since $\mathcal{M}_2$ and $\mathcal{M}_3$ are incomparable in terms of the information transfer from the source, a conservative approach would be to *synthesize* from both mappings a new mapping $\mathcal{M}'$ that only transfers the shared source information that is being mapped by both $\mathcal{M}_2$ and $\mathcal{M}_3$. Since $\mathcal{M}_2$ is transferring the first and the second components of relation A, and $\mathcal{M}_3$ is transferring the first and the third components of relation A, then $\mathcal{M}'$ can be defined as a mapping that transfers only the first component:

$$\mathcal{M}' : A(x, y, z) \rightarrow \exists u\, T(x, u).$$

In our framework, $\mathcal{M}'$ is the *intersection* of $\mathcal{M}_2$ and $\mathcal{M}_3$. Formally, the intersection of $\mathcal{M}_2$ and $\mathcal{M}_3$ is a new mapping $\mathcal{N}$ that transfers less information than $\mathcal{M}_2$ and $\mathcal{M}_3$ and such that any other mapping that transfer less information than $\mathcal{M}_2$ and $\mathcal{M}_3$ transfers no more information than $\mathcal{N}$. That is, the intersection is the *greatest lower bound* with respect to the information transfer order. Note that the actual target schema is not important for the information transfer ordering, and thus the intersection is not unique. For instance, one can show that the mapping $\mathcal{M}_1$ transfers the same information as $\mathcal{M}'$ and therefore also expresses the intersection of $\mathcal{M}_2$ and $\mathcal{M}_3$.

We formalize the notion of intersection and study several of its properties. In the above example, computing the intersection was an easy task, but we show that in general, intersecting mappings is not trivial. In fact, we prove that even

for mappings specified by st-tgds, the intersection may not be expressible in First-Order logic (FO). On the other hand, we prove that Existential Second-Order logic (ESO) suffices to express the intersection of such mappings.

The dual operator is the *union* of schema mappings. Intuitively, the union of two mappings is a new mapping that transfers the *sum* of all the information transferred by both initial mappings. In our example above, the union of $\mathcal{M}_2$ and $\mathcal{M}_3$ is the mapping

$$\mathcal{M}' : A(x, y, z) \rightarrow T(x, y) \wedge S(x, z)$$

Formally, the union of $\mathcal{M}_2$ and $\mathcal{M}_3$ is a new mapping $\mathcal{N}$ that transfers more information than $\mathcal{M}_2$ and $\mathcal{M}_3$ and such that any other mapping that transfers more information than $\mathcal{M}_2$ and $\mathcal{M}_3$ also transfers more information than $\mathcal{N}$. That is, the union is the *least upper bound* with respect to the information transfer order. As for the case of the intersection, dealing with union is not always trivial. For example, one might be tempted to state that the following mapping $\mathcal{M}''$ is also a union of $\mathcal{M}_2$ and $\mathcal{M}_3$:

$$\mathcal{M}'' : A(x, y, z) \rightarrow R(x, y, z)$$

but it can be shown that $\mathcal{M}''$ is strictly more informative than $\mathcal{M}'$ and thus does not define the union for $\mathcal{M}_2$ and $\mathcal{M}_3$. However, if we are given the functional dependencies $\{A[1] \longrightarrow A[2], A[1] \longrightarrow A[3]\}$ over the source schema, then $\mathcal{M}''$ becomes the union of $\mathcal{M}_2$ and $\mathcal{M}_3$. We show that, in the absence of source constraints, the union is considerably easier to handle compared with the intersection. In particular, it can be shown that given mappings specified by a set of st-tgds their union can also be specified by a set of st-tgds.

Organisation of the paper and summary of results. In Section 2, we recall some basic notions and results. A conclusion is given in Section 6. The main results of the paper are detailed in Sections 3 – 5:

- *New operators.* In Section 3, we introduce the union and intersection operators of schema mappings and state our main results, namely: for a large class of mappings (containing the class of st-tgds) the union and intersection always exist. More precisely, this class of mappings is the class Rec of all mappings that have a *maximum recovery* [4] (we recall the definition of maximum recovery in Section 2). Our new operators allow us to define a lattice of the mappings in Rec w.r.t. to the information transfer order, s.t. the union (resp. intersection) corresponds to the least upper bound (resp. greatest lower bound) of two mappings.

- *Existence of the union.* In Section 4, we show for the class Rec that the union of two mappings always exists. The proof is constructive in that we describe how to obtain the union. For mappings defined by a set of st-tgds we show that the union can also be expressed by st-tgds. This allows us to prove an NEXPTIME upper bound for checking if some mapping is the union of two other mappings for the case of st-tgds.

• *Existence of the intersection.* In Section 5, we show several fundamental results on the existence of the intersection. First, for two mappings from the class REC, the intersection always exists. However, in general, even for the restricted case of st-tgds, this intersection is not expressible in First-Order logic (FO). On the other hand, in Existential Second-Order logic (ESO) it is always possible to express the intersection of mappings defined by st-tgds.

2 Preliminaries

2.1 Schemas and schema mappings

A *schema* $\mathbf{S}$ is a finite set $\{R_1, \ldots, R_k\}$ of relation symbols, with each R_i having a fixed arity $n_i \geq 0$. Let $\mathbf{D}$ be a countably infinite domain. An *instance* I of $\mathbf{S}$ assigns to each relation symbol R_i of $\mathbf{S}$ a finite relation $R_i^I \subseteq \mathbf{D}^{n_i}$. $\text{Inst}(\mathbf{S})$ denotes the set of all instances of $\mathbf{S}$. We denote by $\text{dom}(I)$ the set of all elements that occur in any of the relations R_i^I. We say that $R_i(t)$ is a *fact* of I if $t \in R_i^I$. We sometimes denote an instance by its set of facts.

Given schemas $\mathbf{S}$ and $\mathbf{T}$, a *schema mapping* (or just *mapping*) from $\mathbf{S}$ to $\mathbf{T}$ is a subset of $\text{Inst}(\mathbf{S}) \times \text{Inst}(\mathbf{T})$. We say that J is a *solution for I under $\mathcal{M}$* whenever $(I, J) \in \mathcal{M}$. The set of all solutions for I under $\mathcal{M}$ is denoted by $\text{Sol}_{\mathcal{M}}(I)$. For a mapping $\mathcal{M}$ from $\mathbf{S}$ to $\mathbf{T}$, we denote by $\text{dom}(\mathcal{M})$ the set of all instances $I \in \text{Inst}(\mathbf{S})$ such that $\text{Sol}_{\mathcal{M}}(I) \neq \emptyset$. Moreover, $\mathcal{M}$ is said to be *total* if $\text{dom}(\mathcal{M}) = \text{Inst}(\mathbf{S})$.

Notice that mappings are binary relations, and thus one can define the composition of mappings as for the composition of binary relations [15, 7]. Let $\mathcal{M}_{12}$ be a mapping from schema $\mathbf{S}_1$ to schema $\mathbf{S}_2$ and $\mathcal{M}_{23}$ a mapping from $\mathbf{S}_2$ to schema $\mathbf{S}_3$. Then $\mathcal{M}_{12} \circ \mathcal{M}_{23}$ is a mapping from $\mathbf{S}_1$ to $\mathbf{S}_3$ given by the set $\{(I, J) \in \text{Inst}(\mathbf{S}_1) \times \text{Inst}(\mathbf{S}_3) \mid \text{there exists } K \text{ such that } (I, K) \in \mathcal{M}_{12} \text{ and } (K, J) \in \mathcal{M}_{23}\}$ [15, 7].

2.2 Dependencies and definability of mappings

Given disjoint schemas $\mathbf{S}$ and $\mathbf{T}$, a *source-to-target tuple-generating dependency* (st-tgd) from $\mathbf{S}$ to $\mathbf{T}$ is a sentence of the form $\forall \bar{x} \forall \bar{y}(\varphi(\bar{x}, \bar{y}) \rightarrow \exists \bar{z}\ \psi(\bar{x}, \bar{z}))$, where $\varphi(\bar{x}, \bar{y})$ is a conjunctive query (CQ) over $\mathbf{S}$, and $\psi(\bar{x}, \bar{z})$ is a CQ over $\mathbf{T}$. The left-hand side of the implication in an st-tgd is called the *premise*, and the right-hand side the *conclusion*. A *full st-tgd* is an st-tgd with no existentially quantified variables in its conclusion. We usually omit the universal quantifiers when writing st-tgds. Suppose that we are given a set Σ of logical formulas over the schemas $\mathbf{S}$ and $\mathbf{T}$, e.g., a set of st-tgds from $\mathbf{S}$ and $\mathbf{T}$, a set of First-Order formulas or of Existential Second-Order formulas over the schemas $\mathbf{S}$ and $\mathbf{T}$. We say that a mapping $\mathcal{M}$ is *specified* by Σ, denoted by $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, if for every $(I, J) \in \text{Inst}(\mathbf{S}) \times \text{Inst}(\mathbf{T})$, we have $(I, J) \in \mathcal{M}$ if and only if (I, J) satisfies Σ.

As is customary in the data exchange literature, we assume the existence of two disjoint sets of elements: constant values $\mathbf{C}$ and null values $\mathbf{N}$. Thus, for a mapping defined by st-tgds, we assume that source instances are constructed by using only elements from $\mathbf{C}$, while target instances are constructed by using elements from $\mathbf{C} \cup \mathbf{N}$.

2.3 Maximum recovery

The notion of maximum recovery proposed in [4] is fundamental to our study. It provides a natural notion for inverting schema mappings. In [4] the authors first define recoveries of schema mappings and then restrict them to maximum recoveries. Given a mapping $\mathcal{M}$ from $\mathbf{S}_1$ to $\mathbf{S}_2$, we say that $\mathcal{M}'$ from $\mathbf{S}_2$ to $\mathbf{S}_1$ is a *recovery* of $\mathcal{M}$ if for every instance I in $\mathbf{S}_1$, it holds that $(I, I) \in \mathcal{M} \circ \mathcal{M}'$. In symbols, $\mathcal{M}'$ is a recovery of $\mathcal{M}$ if $\mathrm{Id} \subseteq \mathcal{M} \circ \mathcal{M}'$, where Id is the *identity mapping* $\{(I, I) \mid I \in \mathrm{Inst}(\mathbf{S}_1)\}$. Moreover, $\mathcal{M}'$ is said to be a *maximum recovery* of $\mathcal{M}$ if for every other recovery $\mathcal{M}''$ of $\mathcal{M}$, it holds that $\mathcal{M} \circ \mathcal{M}' \subseteq \mathcal{M} \circ \mathcal{M}''$. Intuitively, $\mathcal{M}'$ is a maximum recovery of $\mathcal{M}$ if $\mathcal{M} \circ \mathcal{M}'$ is as close as possible to the identity mapping Id. We write Rec to denote the class of all total mappings that admit a maximum recovery.

2.4 Information Transfer

In [1] a notion of *information transfer* for schema mappings was defined to compare the amount of information that two mappings transfer from source to target. Formally, given mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ with the same source schema $\mathbf{S}$, mapping $\mathcal{M}_1$ *transfers at least as much source information as* $\mathcal{M}_2$, denoted by $\mathcal{M}_2 \preceq_{inf} \mathcal{M}_1$ if there exists a mapping $\mathcal{N}$ such that $\mathcal{M}_1 \circ \mathcal{N} = \mathcal{M}_2$ [1]. That is, $\mathcal{M}_2 \preceq_{inf} \mathcal{M}_1$ if $\mathcal{M}_2$ can be constructed from $\mathcal{M}_1$ via mapping composition. Notice that $\preceq_{inf}$ is a *pre-order*, i.e., $\preceq_{inf}$ is a reflexive and transitive, but not antisymmetric relation. Thus, we say that $\mathcal{M}_1$ and $\mathcal{M}_2$ transfer the same information from the source, denoted by $\mathcal{M}_1 \equiv_{inf} \mathcal{M}_2$ if $\mathcal{M}_1 \preceq_{inf} \mathcal{M}_2$ and $\mathcal{M}_2 \preceq_{inf} \mathcal{M}_1$. By slight abuse of notation we consider $\preceq_{inf}$ as an order (rather than a pre-order) by identifying a mapping $\mathcal{M}$ with the equivalence class of all mappings $\equiv_{inf}$-equivalent with $\mathcal{M}$.

3 The union and intersection operators

Below, we make use of $\preceq_{inf}$ to define the union and the intersection of two mappings. Intuitively the union is a mapping that transfers the *sum* of all the information transferred by the two initial mappings. Analogously, we define the intersection of two mappings as a mapping that transfers only information which is transferred *by each* of the initial mappings.

Definition 1. *Let $\mathcal{C}$ be a class of mappings and $\mathcal{M}_1$ and $\mathcal{M}_2$ two mappings in $\mathcal{C}$ with the same source schema. The* union *of $\mathcal{M}_1$ and $\mathcal{M}_2$ w.r.t. $\mathcal{C}$, is a mapping $\mathcal{M} \in \mathcal{C}$ such that:*

1. *$\mathcal{M}_1 \preceq_{inf} \mathcal{M}$,*
2. *$\mathcal{M}_2 \preceq_{inf} \mathcal{M}$, and*
3. *if $\mathcal{N}$ is a mapping in $\mathcal{C}$ with $\mathcal{M}_1 \preceq_{inf} \mathcal{N}$ and $\mathcal{M}_2 \preceq_{inf} \mathcal{N}$, then $\mathcal{M} \preceq_{inf} \mathcal{N}$.*

The union of $\mathcal{M}_1$ and $\mathcal{M}_2$ w.r.t. $\mathcal{C}$ is denoted by $\mathcal{M}_1 \sqcup_{\mathcal{C}} \mathcal{M}_2$.

Definition 2. *Let $\mathcal{C}$ be a class of mappings and $\mathcal{M}_1$ and $\mathcal{M}_2$ two mappings in $\mathcal{C}$ with the same source schema. The* intersection *of $\mathcal{M}_1$ and $\mathcal{M}_2$ w.r.t. $\mathcal{C}$, is a mapping $\mathcal{M} \in \mathcal{C}$ such that:*

1. $\mathcal{M} \preceq_{inf} \mathcal{M}_1$,
2. $\mathcal{M} \preceq_{inf} \mathcal{M}_2$, *and*
3. *if* $\mathcal{N}$ *is a mapping in* $\mathcal{C}$ *with* $\mathcal{N} \preceq_{inf} \mathcal{M}_1$ *and* $\mathcal{N} \preceq_{inf} \mathcal{M}_2$, *then* $\mathcal{N} \preceq_{inf} \mathcal{M}$.

The intersection of $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *is denoted by* $\mathcal{M}_1 \sqcap_{\mathcal{C}} \mathcal{M}_2$.

When the class $\mathcal{C}$ is clear from the context, we just write $\mathcal{M}_1 \sqcup \mathcal{M}_2$ (resp. $\mathcal{M}_1 \sqcap \mathcal{M}_2$) for the union (resp. for the intersection) of two mappings. Notice that the definition of the union of mappings is just the *least upper bound* (supremum) of $\mathcal{M}_1$ and $\mathcal{M}_2$ w.r.t. $\preceq_{inf}$ (inside the class of mappings $\mathcal{C}$). Analogously, the intersection of mappings is just the *greatest lower bound* (infimum) of $\mathcal{M}_1$ and $\mathcal{M}_2$ w.r.t. $\preceq_{inf}$ (inside the class $\mathcal{C}$). Also notice that the union and the intersection as defined above are unique up to the equivalence relation $\equiv_{inf}$. This is why we speak of *the* union resp. intersection of two mappings.

Notice that with the definition of union and intersection based on the order $\preceq_{inf}$ it is by no means evident that for any two mappings the union or intersection always exists. Thus, a first important question that needs to be answered for these operators is for which classes of mappings the existence of the intersection or the union is guaranteed. As we will show next, the class Rec of mappings having a maximum recovery will play a fundamental role in determining the existence of the union and intersection.

Beside existence, there are two other important questions regarding these operators that need to be addressed. One is the question of expressiveness: what is the mapping language needed to express the union/intersection when it exists? Another main question is about computing these operators: is there an algorithm to compute the union/intersection? One of our main results is the following general result that gives a positive answer to the existence question.

Theorem 1. *There exists a class* $\mathcal{R}$ *of mappings (that contains the class of mappings specified by st-tgds), such that for every pair of mappings* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *in* $\mathcal{R}$ *having the same source schema, the union* $\mathcal{M}_1 \sqcup_{\mathcal{R}} \mathcal{M}_2$ *and the intersection* $\mathcal{M}_1 \sqcap_{\mathcal{R}} \mathcal{M}_2$ *always exist.*

We will show that Rec is such a class $\mathcal{R}$ that satisfies the statement of Theorem 1. By using notions of lattice theory, Theorem 1 can be restated as follows. Recall that given an order relation $\leq$, a lattice is a structure $\langle \mathcal{A}, \leq \rangle$ such that every two elements $X, Y \in \mathcal{A}$ have a least upper bound (supremum) and a greatest lower bound (infimum) in $\mathcal{A}$. Then Theorem 1 can be formulated as follows.

Theorem 2. *Let* $\mathbf{S}$ *be a relational schema. There exists a class* $\mathcal{R}_{\mathbf{S}}$ *that contains the class of all mappings specified by st-tgds having* $\mathbf{S}$ *as source schema, such that* $\langle \mathcal{R}_{\mathbf{S}}, \preceq_{inf} \rangle$ *is a lattice (up to* $\equiv_{inf}$*-equivalence).*

Theorem 1 is proved by combining the results in the following sections for union and intersection. Theorem 2 is an immediate consequence of Theorem 1 given the following proposition:

Proposition 1. *The union and intersection of mappings are invariant under $\equiv_{inf}$-equivalence. Formally, let $\mathcal{M}_1, \mathcal{M}'_1, \mathcal{M}_2$, and $\mathcal{M}'_2$ be mappings from some class $\mathcal{C}$ with $\mathcal{M}_1 \equiv_{inf} \mathcal{M}'_1$ and $\mathcal{M}_2 \equiv_{inf} \mathcal{M}'_2$. Then the following relations hold:*
(1) If $\mathcal{M}_1 \sqcup_{\mathcal{C}} \mathcal{M}_2$ exists, then $\mathcal{M}'_1 \sqcup_{\mathcal{C}} \mathcal{M}'_2$ exists as well and the equivalence $\mathcal{M}_1 \sqcup_{\mathcal{C}} \mathcal{M}_2 \equiv_{inf} \mathcal{M}'_1 \sqcup_{\mathcal{C}} \mathcal{M}'_2$ holds.
(2) If $\mathcal{M}_1 \sqcap_{\mathcal{C}} \mathcal{M}_2$ exists, then $\mathcal{M}'_1 \sqcap_{\mathcal{C}} \mathcal{M}'_2$ exists as well and the equivalence $\mathcal{M}_1 \sqcap_{\mathcal{C}} \mathcal{M}_2 \equiv_{inf} \mathcal{M}'_1 \sqcap_{\mathcal{C}} \mathcal{M}'_2$ holds.

Intuitively, Proposition 1 states that union and intersection of mappings are preserved under $\equiv_{inf}$-equivalence. The proposition follows immediately from the definition of $\sqcup_{\mathcal{C}}$ and $\sqcap_{\mathcal{C}}$.

4 Existence of the union

In this section we propose a straightforward method to compute the union of mappings specified by st-tgds (w.r.t. the class of mappings specified by st-tgds). This method will allow us to provide a positive answer to all of the questions concerning the existence, expressiveness, and computation of the union for this class of mappings. In contrast, we will show in Section 5 that dealing with the intersection operator is considerably more difficult.

The procedure to compute the union is very simple. Let $\mathcal{M}_1 = (\mathbf{S}, \mathbf{T}_1, \Sigma_1)$ and $\mathcal{M}_2 = (\mathbf{S}, \mathbf{T}_2, \Sigma_2)$ be two mappings specified by st-tgds. Let $\widehat{\mathbf{T}}_2$ be a copy of $\mathbf{T}_2$ such that $\widehat{\mathbf{T}}_2$ is disjoint with $\mathbf{T}_1$, and let $\widehat{\Sigma}_2$ be the set of dependencies that results from Σ_2 by replacing every relation name in $\mathbf{T}_2$ by its copy in $\widehat{\mathbf{T}}_2$. Consider the mapping $\mathcal{M}' = (\mathbf{S}, \mathbf{T}_1 \cup \widehat{\mathbf{T}}_2, \Sigma_1 \cup \widehat{\Sigma}_2)$. Then $\mathcal{M}'$ is the union of $\mathcal{M}_1$ and $\mathcal{M}_2$. From this we obtain the following result.

Proposition 2. *Let $\mathcal{S}$ be the class of mappings specified by st-tgds, and $\mathcal{M}_1 = (\mathbf{S}, \mathbf{T}_1, \Sigma_1)$ and $\mathcal{M}_2 = (\mathbf{S}, \mathbf{T}_2, \Sigma_2)$ be mappings such that Σ_1 and Σ_2 are sets of st-tgds. Then the union $\mathcal{M}_1 \sqcup_{\mathcal{S}} \mathcal{M}_2$ always exists. Moreover, there exists an algorithm which, given $\mathcal{M}_1$ and $\mathcal{M}_2$, computes $\mathcal{M}_1 \sqcup_{\mathcal{S}} \mathcal{M}_2$ in polynomial time.*

Proposition 2 follows from a more general result on the existence of the union for a class of mappings that properly contains the class of mappings specified by st-tgds. Recall from Section 2.3 that we write Rec to denote the class of all total mappings that admit a maximum recovery. It was shown in [4] that every mapping specified by st-tgds is total and has a maximum recovery, and thus Rec contains the class of all mappings specified by st-tgds. The following is the general result for the union operator w.r.t. the class Rec.

Proposition 3. *For every pair of mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ in* Rec *having the same source schema, the union $\mathcal{M}_1 \sqcup_{\text{Rec}} \mathcal{M}_2$ exists.*

Proof (sketch). Let $\mathbf{T}_1$ and $\mathbf{T}_2$ be disjoint schemas, $\mathcal{M}_1$ a mapping in Rec from $\mathbf{S}$ to $\mathbf{T}_1$, and $\mathcal{M}_2$ a mapping in Rec from $\mathbf{S}$ to $\mathbf{T}_2$. Consider the mapping $\mathcal{M}_1 \oplus \mathcal{M}_2$ from $\mathbf{S}$ to $\mathbf{T}_1 \cup \mathbf{T}_2$ defined as follows:

$$\mathcal{M}_1 \oplus \mathcal{M}_2 = \{(I, J_1 \cup J_2) \mid (I, J_1) \in \mathcal{M}_1 \text{ and } (I, J_2) \in \mathcal{M}_2\}.$$

It can be shown that $\mathcal{M}_1 \oplus \mathcal{M}_2$ is the union of $\mathcal{M}_1$ and $\mathcal{M}_2$ w.r.t. Rec. If $\mathbf{T}_1$ and $\mathbf{T}_2$ are not disjoint, one can always construct a copy $\widehat{\mathbf{T}}_2$ of $\mathbf{T}_2$ that is disjoint with $\mathbf{T}_1$, and a mapping $\widehat{\mathcal{M}}_2$ from $\mathbf{S}$ to $\widehat{\mathbf{T}}_2$ such that $\mathcal{M}_2 \equiv_{inf} \widehat{\mathcal{M}}_2$, and then $\mathcal{M}_1 \oplus \widehat{\mathcal{M}}_2$ is the desired union. □

The proof of Proposition 2 follows from the proof of Proposition 3 plus the fact that if $\mathcal{M}_1$ and $\mathcal{M}_2$ are specified by st-tgds, then $\mathcal{M}_1 \oplus \mathcal{M}_2$ can also be specified by a set of st-tgds. The following example shows that computing the union is extremely easy for the case of mappings specified by st-tgds.

Example 1. Let mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ be defined by the following sets of st-tgds: $\mathcal{M}_1 = \{S(x,y) \to T(x,y)\}$; $M_2 = \{S(x,y) \to T(x,y),\ Q(x) \to T(x,x)\}$. The union $\mathcal{M}_1 \sqcup_{\text{Rec}} \mathcal{M}_2$ is simply the mapping $\mathcal{M}$ containing all three dependencies, with appropriately renamed target relation symbols:

$$\mathcal{M} = \{S(x,y) \to T(x,y),\ S(x,y) \to T'(x,y),\ Q(x) \to T'(x,x)\}$$

Proposition 2 also allows us to prove positive algorithmic results regarding the union of schema mappings. The following result follows directly from Proposition 2 and the results in [1] regarding the order $\preceq_{inf}$.

Proposition 4. *Given mappings $\mathcal{M}_1$, $\mathcal{M}_2$, and $\mathcal{M}_3$ specified by st-tgds, it is decidable (in* NEXPTIME*) whether $\mathcal{M}_3$ is the union of $\mathcal{M}_1$ and $\mathcal{M}_2$.*

A legitimate question, of course, is if the characterization in the proof of Proposition 3 also works outside Rec. We have to leave this as an open question for future research. The following proposition shows that this is a tricky problem which may even require some adaptation of the $\preceq_{inf}$ relation.

Proposition 5. *There exists a mapping $\mathcal{M}$ such that $\mathcal{M} \prec_{inf} \mathcal{M} \oplus \mathcal{M}$ (that is $\mathcal{M} \preceq_{inf} \mathcal{M} \oplus \mathcal{M}$ and $\mathcal{M} \oplus \mathcal{M} \not\preceq_{inf} \mathcal{M}$).*

In other words, the $\preceq_{inf}$ order displays an unexpected behaviour for mappings outside Rec: intuitively, one would expect that the amount of source information transferred remains unchanged if, for every source instance I, we combine all pairs of solutions of I. For mappings in Rec, this is of course the case. In contrast, by Proposition 5, there are mappings outside Rec such that the amount of source information transferred strictly increases by this simple syntactic trick.

5 Existence of the intersection

In this section we study the existence of the intersection. The main result is stated in the following theorem. Again, we use Rec to denote the class of total mappings that have a maximum recovery.

Theorem 3. *For every pair of mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ in* Rec *having the same source schema, the intersection $\mathcal{M}_1 \sqcap_{\text{Rec}} \mathcal{M}_2$ exists.*

Proof (sketch). To describe the proof of the theorem we need to introduce some technical notions. Let $\mathbf{S}$ be a schema and consider a mapping $\mathcal{M}$ from $\mathbf{S}$ to $\mathbf{S}$ (that is $\mathcal{M} \subseteq \text{Inst}(\mathbf{S}) \times \text{Inst}(\mathbf{S})$). For a positive integer k, we define $\mathcal{M}^k$ recursively as follows:

$$\mathcal{M}^1 = \mathcal{M},$$
$$\mathcal{M}^{k+1} = \mathcal{M} \circ \mathcal{M}^k.$$

We shall also define $\mathcal{M}^+$ as the following mapping from $\mathbf{S}$ to $\mathbf{S}$:

$$\mathcal{M}^+ = \bigcup_{i=1}^{\infty} \mathcal{M}^i.$$

Notice that $\mathcal{M}^+$ is the *transitive closure* of $\mathcal{M}$ when it is viewed as a binary relation over $\text{Inst}(\mathbf{S})$.

Now consider mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ in the statement of the theorem. Given that $\mathcal{M}_1$ and $\mathcal{M}_2$ are mappings in REC, we know that there exist mappings $\mathcal{M}_1'$ and $\mathcal{M}_2'$ such that $\mathcal{M}_1'$ is a maximum recovery of $\mathcal{M}_1$, and $\mathcal{M}_2'$ is a maximum recovery of $\mathcal{M}_2$. Now consider the mapping $\mathcal{M}$ given by

$$\mathcal{M} \;=\; \Big((\mathcal{M}_1 \circ \mathcal{M}_1') \cup (\mathcal{M}_2 \circ \mathcal{M}_2') \Big)^+.$$

It can be shown that $\mathcal{M}$ is the intersection $\mathcal{M}_1 \sqcap_{\text{REC}} \mathcal{M}_2$. □

Since every mapping given by st-tgds is total and has a maximum recovery [4], from Theorem 3 we obtain that for mappings specified by st-tgds the intersection (w.r.t. the class REC) always exists.

Notice that Theorem 3 is only about existence and says nothing about the language needed to express the intersection of mappings specified by st-tgds. The following result shows that as opposed to the case of the union operator, the intersection of mappings specified by st-tgds may not be expressible in First-Order logic (FO).

Theorem 4. *There exist mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ specified by st-tgds such that $\mathcal{M}_1 \sqcap_{\text{REC}} \mathcal{M}_2$ cannot be specified by a set of* FO *sentences.*

Proof (sketch). Consider the schemas $\mathbf{S} = \{A(\cdot,\cdot), B(\cdot,\cdot)\}$, $\mathbf{T}_1 = \{T_1(\cdot,\cdot)\}$, and $\mathbf{T}_2 = \{T_2(\cdot,\cdot)\}$. In the proof we use mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ from $\mathbf{S}$ to $\mathbf{T}_1$ and from $\mathbf{S}$ to $\mathbf{T}_2$, respectively, specified by the following st-tgds:

$$\mathcal{M}_1: \quad \exists u\, (A(x,u) \wedge B(u,y)) \rightarrow T_1(x,y)$$
$$\mathcal{M}_2: \quad \exists u\, (B(x,u) \wedge A(u,y)) \rightarrow T_2(x,y)$$

It can be shown by an argument based on Ehrenfeucht-Fraïssé games that $\mathcal{M}_1 \sqcap_{\text{REC}} \mathcal{M}_2$ cannot be expressed by an FO sentence. □

Notice that the above theorem states that $\mathcal{M}_1 \sqcap_{\text{REC}} \mathcal{M}_2$ is not expressible in FO. In principle, it might be the case that if we restrict ourselves to the intersection with respect to a smaller class of mappings, for example the class of mappings specified by st-tgds, then we could obtain better expressibility results. The following proposition shows a negative result in this respect. This is a corollary of the proof of Theorem 4.

Proposition 6. *Let $\mathcal{S}$ be the class of mappings specified by st-tgds. Then there are mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ in $\mathcal{S}$ such that $\mathcal{M}_1 \sqcap_{\mathcal{S}} \mathcal{M}_2$ does not exist.*

This raises the question as to which language is expressive enough to specify the intersection of two mappings specified by st-tgds. Below we provide an answer to this question.

Theorem 5. *Given two schema mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ given by st-tgds, the intersection $\mathcal{M}_1 \sqcap_{\text{REC}} \mathcal{M}_2$ is expressible by an Existential Second-Order logic (ESO) formula.*

Proof (sketch). Let $\mathbf{S}$ be the source schema of $\mathcal{M}_1$ and $\mathcal{M}_2$, and $\widehat{\mathbf{S}}$ a copy of schema $\mathbf{S}$. Moreover, let $\mathcal{M}_1'$ and $\mathcal{M}_2'$ be respective maximum recoveries of $\mathcal{M}_1$ and $\mathcal{M}_2$. One can show that the composition $\mathcal{M}_1 \circ \mathcal{M}_1'$ can be expressed as an FO formula:

$$\forall \boldsymbol{x}_1(\varphi_1(\boldsymbol{x}_1) \rightarrow \psi_1(\boldsymbol{x}_1)) \wedge \ldots \wedge \forall \boldsymbol{x}_n(\varphi_n(\boldsymbol{x}_n) \rightarrow \psi_n(\boldsymbol{x}_n))$$

where $\varphi_i(\boldsymbol{x}_i)$ and $\psi_i(\boldsymbol{x}_i)$ are FO formulas over $\mathbf{S}$ and $\widehat{\mathbf{S}}$, respectively. Similarly, $\mathcal{M}_2 \circ \mathcal{M}_2'$ can be expressed as $\forall \boldsymbol{y}_1(\alpha_1(\boldsymbol{y}_1) \rightarrow \beta_1(\boldsymbol{y}_1)) \wedge \ldots \wedge \forall \boldsymbol{y}_m(\alpha_m(\boldsymbol{y}_m) \rightarrow \beta_m(\boldsymbol{y}_m))$. The formula representing the intersection is based on the construction in the proof of Theorem 3, thus we need to show how to express the transitive closure of $\mathcal{M}_1 \circ \mathcal{M}_1' \cup \mathcal{M}_2 \circ \mathcal{M}_2'$. For this we use an intermediate schema $\widetilde{\mathbf{S}}$ constructed as follows: for every n-ary relation R of $\mathbf{S}$, we include an $(n+1)$-ary relation $\widetilde{R}$ in $\widetilde{\mathbf{S}}$. The idea is that an atom $\widetilde{R}(\boldsymbol{a}, g)$ will represent the atom $R(\boldsymbol{a})$ in the generation g of the computation of the transitive closure. Now, to define the intersection, we use an ESO formula of the form

$$\exists \widetilde{\mathbf{S}}\ \exists s\ \exists \text{zero}\ (\Omega_s \wedge \Omega_{\widetilde{R}} \wedge \Omega_{\widetilde{R}}^{E})$$

where $\exists \widetilde{\mathbf{S}}$ denotes an existential quantification over all relation symbols in $\widetilde{\mathbf{S}}$, s is a function symbol, and zero a first order variable. The rest of the formulas is constructed as follows: Ω_s is the formula $\forall x \forall y\big((s(x) = s(y) \rightarrow x = y) \wedge \neg(s(x) = x) \wedge \neg(s(x) = \text{zero})\big)$ that defines a successor function, with zero as the first element; $\Omega_{\widetilde{R}}$ corresponds to the following FO formula (we assume $\mathbf{S} = \{R_1, \ldots, R_k\}$ and $\boldsymbol{z}_i$ is a tuple of variables of the same arity as R_i):

$$\big(\forall \boldsymbol{z}_1(R_1(\boldsymbol{z}_1) \rightarrow \widetilde{R}_1(\boldsymbol{z}_1, \text{zero})) \wedge \cdots \wedge \forall \boldsymbol{z}_k(R_k(\boldsymbol{z_k}) \rightarrow \widetilde{R}_k(\boldsymbol{z}_k, \text{zero}))\big) \wedge$$
$$\forall g\Big(\bigwedge_{i=1}^{n} \forall \boldsymbol{x}_i\big[\widetilde{\varphi}_i(\boldsymbol{x}_i, g) \rightarrow \widetilde{\psi}_i(\boldsymbol{x_i}, s(g))\big] \ \vee \ \bigwedge_{i=1}^{m} \forall \boldsymbol{y}_i\big[\widetilde{\alpha}_i(\boldsymbol{y}_i, g) \rightarrow \widetilde{\beta}_i(\boldsymbol{y_i}, s(g))\big] \Big)$$

where $\widetilde{\varphi}_i(\boldsymbol{x}_i, g)$ is obtained from $\varphi_i(\boldsymbol{x}_i)$ by replacing every relational symbol $R(\boldsymbol{z})$ by $\widetilde{R}(\boldsymbol{z}, g)$, and $\widetilde{\psi}_i(\boldsymbol{x_i}, s(g))$ is obtained from $\psi_i(\boldsymbol{x_i})$ by replacing every relational symbol $\widehat{R}(\boldsymbol{z})$ by $\widetilde{R}(\boldsymbol{z}, s(g))$, and similarly for $\widetilde{\alpha}_i$ and $\widetilde{\beta}_i$. The intuition is that the first line initializes the relations $\widetilde{R}_i$ at generation 0, and the second line mimics a formula representing $(\mathcal{M}_1 \circ \mathcal{M}_1' \cup \mathcal{M}_2 \circ \mathcal{M}_2')^+$ over schema $\widetilde{\mathbf{S}}$. Finally, $\Omega^E_{\widetilde{R}}$ just extracts the target relations at some generation g of the transitive closure:

$$\exists g\big(\forall \boldsymbol{z}_1(\widetilde{R}_1(\boldsymbol{z}_1, g) \to \widehat{R}_1(\boldsymbol{z}_1)) \ \wedge \cdots \wedge \ \forall \boldsymbol{z}_k(\widetilde{R}_k(\boldsymbol{z}_k, g) \to \widehat{R}_k(\boldsymbol{z}_k))\big). \qquad \square$$

Example 2. Recall the mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ from Example 1. Composed with their maximum recoveries, they have the following form (see [4]):

$$\begin{aligned}\mathcal{M}_1 \circ \mathcal{M}_1' &= \{S(x_1, x_2) \to \widehat{S}(x_1, x_2)\} \text{ and}\\ \mathcal{M}_2 \circ \mathcal{M}_2' &= \{S(x_1, x_2) \to \widehat{S}(x_1, x_2) \vee (x_1 = x_2 \wedge \widehat{Q}(x_1)),\\ &\qquad\quad Q(x_1) \to \widehat{S}(x_1, x_1) \vee \widehat{Q}(x_1)\}.\end{aligned}$$

The intersection $\mathcal{M}_1 \sqcap_{\text{REC}} \mathcal{M}_2$ is expressed by the following ESO formula:

$$\begin{aligned}\exists \widetilde{S} \exists \widetilde{Q} \exists s \exists \,\text{zero} \Big(& \forall x \forall y\big((s(x) = s(y) \to x = y) \wedge s(x) \neq x \wedge s(x) \neq \text{zero}\big) \wedge\\ & \forall x_1 \forall x_2\big(S(x_1, x_2) \to \widetilde{S}(x_1, x_2, \text{zero})\big) \wedge \forall x\big(Q(x) \to \widetilde{Q}(x, \text{zero})\big) \wedge\\ \forall g \Big(& \forall x_1 \forall x_2\big(\widetilde{S}(x_1, x_2, g) \to \widetilde{S}(x_1, x_2, s(g))\big) \vee\\ & \Big[\forall x_1 \forall x_2\big(\widetilde{S}(x_1, x_2, g) \to \widetilde{S}(x_1, x_2, s(g)) \vee (x_1 = x_2 \wedge \widetilde{Q}(x_1, s(g)))\big) \wedge\\ & \ \forall x_1 \forall x_2\big(\widetilde{Q}(x_1, g) \to \widetilde{S}(x_1, x_1, s(g)) \vee \widetilde{Q}(x_1, s(g))\big)\Big]\Big) \wedge\\ \exists g' \ \big(& \forall x_1 \forall x_2(\widetilde{S}(x_1, x_2, g') \to \widehat{S}(x_1, x_2)) \wedge \forall x(\widetilde{Q}(x, g') \to \widehat{Q}(x))\big)\Big)\end{aligned}$$

6 Conclusion

In this work, we have introduced two new operators *union* and *intersection* on schema mappings. We have proved that these operators allow us to define a lattice w.r.t. the order $\preceq_{inf}$ (up to $\equiv_{inf}$-equivalence) for the mappings in REC (i.e., mappings having a maximum recovery). In particular, we have shown that the union and intersection always exist for mappings in REC. When restricting us to the simple case of mappings specified by st-tgds it has turned out that the intersection operator is considerably more difficult to handle than the union operator. More specifically, the union of two mappings specified by st-tgds can again be specified by a set of st-tgds. In contrast, First-Order logic (FO) does in general not suffice to express the intersection of such mappings.

A lot of interesting research questions have been left for future work. First of all, while our definitions of $\sqcup$ and $\sqcap$ are applicable to arbitrary mappings, we have restricted ourselves to the mappings in REC for investigating the questions of existence, expressiveness, and computability of $\sqcup$ and $\sqcap$. We would like to extend

this study to arbitrary mappings. As has been illustrated in Proposition 5, such an extension may even require an adaptation of the $\preceq_{inf}$-relation.

Recall that in Theorem 5 we have shown that ESO is expressive enough to specify the intersection of two mappings given by sets of st-tgds. Further analysis is required to determine if a smaller fragment of ESO would also suffice. In addition it would be interesting to identify restrictions on the st-tgds so that the intersection of mappings specified by such st-tgds is FO-expressible.

We would also like to extend our study to further set operators on schema mappings. Above all, we would like to study the complement of a mapping $\mathcal{M}$ (i.e., a mapping that transfers all the source information *not* transferred by $\mathcal{M}$) and, more generally, the set difference of two mappings $\mathcal{M}$ and $\mathcal{N}$ (i.e., a mapping that transfers all the source information that is transferred by $\mathcal{M}$ but not by $\mathcal{N}$). Overall, we think that the union and intersection operators can be crucial in not only defining operators such as difference and complement, but in laying the foundation to a framework of similar operators on schema mappings.

Given mappings from a source schema $\mathbf{S}$ to a target schema $\mathbf{T}$, an interesting question is if the result of the union or intersection can be also specified as a mapping from $\mathbf{S}$ to $\mathbf{T}$. Our definitions do not force the same target schema to be used when constructing a union or intersection, but we consider this setting to be the most common in practice. We leave the investigation of this case as a yet another item in the future research agenda.

Acknowledgements This work has been funded in part by Marie Curie action IRSES under Grant No. 24761 (Net2), and by the Vienna Science and Technology Fund (WWTF) through project ICT08-032. Jorge Pérez has been supported by Fondecyt grant 11110404 and by VID grant U-Inicia 11/04 Universidad de Chile.

References

1. M. Arenas, J. Pérez, J. L. Reutter, and C. Riveros. Foundations of schema mapping management. In *Proc. PODS*, pages 227–238, 2010.
2. Marcelo Arenas, Ronald Fagin, and Alan Nash. Composition with target constraints. In *Proc. ICDT'10*, ACM International Conference Proceeding Series, pages 129–142. ACM, 2010.
3. Marcelo Arenas, Jorge Pérez, Juan L. Reutter, and Cristian Riveros. Composition and inversion of schema mappings. *SIGMOD Record*, 38(3):17–28, 2009.
4. Marcelo Arenas, Jorge Pérez, and Cristian Riveros. The recovery of a schema mapping: Bringing exchanged data back. *ACM Trans. Database Syst.*, 34(4), 2009.
5. P. Bernstein. Applying model management to classical meta data problems. In *Proc. CIDR*, 2003.
6. P. Bernstein and S. Melnik. Model management 2.0: manipulating richer mappings. In *Proc. SIGMOD*, pages 1–12, 2007.
7. R. Fagin, P. G. Kolaitis, L. Popa, and W.-C. Tan. Composing schema mappings: Second-order dependencies to the rescue. *TODS*, 30(4):994–1055, 2005.
8. Ronald Fagin. Inverting schema mappings. *ACM Trans. Database Syst.*, 32(4), 2007.
9. Ronald Fagin, Phokion G. Kolaitis, Renée J. Miller, and Lucian Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

10. Ronald Fagin, Phokion G. Kolaitis, Lucian Popa, and Wang Chiew Tan. Reverse data exchange: Coping with nulls. *ACM Trans. Database Syst.*, 36(2):11, 2011.
11. L. M. Haas, M. A. Hernández, H. Ho, L. Popa, and M. Roth. Clio grows up: from research prototype to industrial tool. In *Proc. SIGMOD*, pages 805–810, 2005.
12. Maurizio Lenzerini. Data integration: A theoretical perspective. In *Proc. PODS*, pages 233–246. ACM, 2002.
13. Leonid Libkin. *Elements of Finite Model Theory*. Springer, 2004.
14. Jayant Madhavan and Alon Y. Halevy. Composing mappings among data sources. In *Proc. VLDB'03*, pages 572–583, 2003.
15. Sergey Melnik. *Generic Model Management: concepts and Algorithms*, volume 2967 of *LNCS*. Springer, 2004.
16. Alan Nash, Philip A. Bernstein, and Sergey Melnik. Composition of mappings given by embedded dependencies. *ACM Trans. Database Syst.*, 32(1):4, 2007.
17. Jorge Pérez. *Schema Mapping Management in Data Exchange Systems*. PhD thesis, Escuela de Ingeniería, Pontificia Universidad Católica de Chile, 2011.