TECHNICAL REPORT

INSTITUT FÜR INFORMATIONSSYSTEME

ABTEILUNG DATENBANKEN UND ARTIFICIAL INTELLIGENCE

Normalization and Optimization of Schema Mappings[*]

DBAI-TR-2011-69

Georg Gottlob, Reinhard Pichler, Vadim Savenkov

[*] *This is an author's archive copy, with complete proofs, of an article published in The VLDB Journal, Vol. 20(2), April 2011, pp. 277–302*

DBAI TECHNICAL REPORT
2011

Institut für Informationssysteme
Abteilung Datenbanken und
Artificial Intelligence
Technische Universität Wien
Favoritenstr. 9
A-1040 Vienna, Austria
Tel: +43-1-58801-18404
Fax: +43-1-58801-18493
sekret@dbai.tuwien.ac.at
www.dbai.tuwien.ac.at

DBAI TECHNICAL REPORT

DBAI TECHNICAL REPORT DBAI-TR-2011-69, 2011

Normalization and Optimization of Schema Mappings

Georg Gottlob [1], Reinhard Pichler, Vadim Savenkov [2]

Abstract. Schema mappings are high-level specifications that describe the relationship between database schemas. They are an important tool in several areas of database research, notably in data integration and data exchange. However, a concrete theory of schema mapping optimization including the formulation of optimality criteria and the construction of algorithms for computing optimal schema mappings is completely lacking to date. The goal of this work is to fill this gap. We start by presenting a system of rewrite rules to minimize sets of source-to-target tuple-generating dependencies. Moreover, we show that the result of this minimization is unique up to variable renaming. Hence, our optimization also yields a schema mapping normalization. By appropriately extending our rewrite rule system, we also provide a normalization of schema mappings containing equality-generating target dependencies. An important application of such a normalization is in the area of defining the semantics of query answering in data exchange, since several definitions in this area depend on the concrete syntactic representation of the mappings. This is, in particular, the case for queries with negated atoms and for aggregate queries. The normalization of schema mappings allows us to eliminate the effect of the concrete syntactic representation of the mapping from the semantics of query answering. We discuss in detail how our results can be fruitfully applied to aggregate queries.

[1] Computing Laboratory, Oxford University, UK georg.gottlob@comlab.ox.ac.uk
[2] Technische Universität Wien, {pichler|savenkov}@dbai.tuwien.ac.at

Acknowledgements: R. Pichler and V. Savenkov acknowledge support by the Vienna Science and Technology Fund (WWTF), project ICT08-032. V. Savenkov is a holder of a scholarship from the European program “Erasmus Mundus External Cooperation Window”. G. Gottlob’s work was supported by EPSRC grant EP/E010865/1 “Schema Mappings and Automated Services for Data Integration and Exchange”. G. Gottlob is the holder of a Royal Society Wolfson Research Merit Award.

1 Introduction

Schema mappings are high-level specifications that describe the relationship between two database schemas. They play an important role in data integration [13,18] and data exchange [9]. A schema mapping is usually given in the form $\mathcal{M} = \langle \mathbf{S}, \mathbf{T}, \Sigma \rangle$, indicating the two database schemas $\mathbf{S}$ and $\mathbf{T}$ plus a set Σ of dependencies. These dependencies express conditions that instances of $\mathbf{S}$ and $\mathbf{T}$ must fulfill. In data exchange, $\mathbf{S}$ and $\mathbf{T}$ are referred to as source and target schema. The dependencies Σ specify, given a source instance (i.e., an instance of $\mathbf{S}$), what a legal target instance (i.e., an instance of $\mathbf{T}$) may look like. Similarly, in data integration, a schema mapping $\mathcal{M}$ describes the relationship between a local data source and a global mediated schema.

Over the past years, schema mappings have been extensively studied (see [17,6] for numerous pointers to the literature). However, only recently, the question of *schema mapping optimization* has been raised. In [10], the foundation for optimization has been laid by defining various forms of equivalence of schema mappings and by proving important properties of the resulting notions. However, a concrete theory of schema mapping optimization including the formulation of optimality criteria and the construction of algorithms for computing optimal schema mappings is completely lacking to date. The goal of this work is to fill this gap. Below, we illustrate the basic ideas of our approach by a series of simple examples, where it is clear "at a glance" what the optimal form of the schema mappings should look like. In fact, one would expect that a human user designs these mappings in their optimal form right from the beginning. However, as more and more progress is made in the area of automatic generation and processing of schema mappings [6,5] we shall have to deal with schema mappings of ever increasing complexity. The optimality of these automatically derived schema mappings is by no means guaranteed and schema mapping optimization will become a real necessity.

For the most common form of schema mappings considered in the literature, the dependencies in Σ are source-to-target tuple-generating dependencies (or s-t tgds, for short) of the form $\forall \mathbf{x}\,(\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y}))$, where the antecedent φ is a conjunctive query (CQ) over $\mathbf{S}$ and the conclusion ψ is a CQ over $\mathbf{T}$. The universal quantification is usually not denoted explicitly. Instead, it is assumed implicitly for all variables in $\varphi(\mathbf{x})$.

Example 1 Consider a schema mapping $\mathcal{M} = \langle \mathbf{S}, \mathbf{T}, \Sigma \rangle$ with $\mathbf{S} = \{L(\cdot,\cdot,\cdot), P(\cdot,\cdot)\}$ and $\mathbf{T} = \{C(\cdot,\cdot)\}$, where L, P, and C are abbreviations for the relational schemas Lecture(title, year, prof), Prof(name, area), and Course (title, prof-area), respectively. Moreover, suppose that Σ consists of two rules expressing the following constraints: If any lecture is specified in the source instance, then the title of all lectures for 3^{rd} year students as well as the area of the professor giving this lecture should be present in the Course-relation of the target instance. Moreover, Σ contains a specific rule which takes care of the lectures given by professors from the database area. We get the following set Σ of s-t tgds:

$$L(x_1, x_2, x_3) \wedge L(x_4, 3, x_5) \wedge P(x_5, x_6) \rightarrow C(x_4, x_6)$$
$$L(x_1, 3, x_2) \wedge P(x_2, \text{'db'}) \rightarrow C(x_1, \text{'db'}) \qquad \square$$

The above schema mapping has a specific form called GAV (global-as-view) [18], i.e., we only have s-t tgds $\varphi(\mathbf{x}) \rightarrow A(\mathbf{x})$, where the conclusion is a single atom $A(\mathbf{x})$ without existentially quantified variables. In this special case, we see a close relationship of schema mappings with unions of conjunctive queries (UCQs). Indeed, given a source instance I over $\mathbf{S}$, the tuples which have to be present in any legal target instance J according to the above schema mapping $\mathcal{M}$ are precisely the tuples in the result of the following UCQ:

$$\begin{array}{ll} ans(x_4, x_6) & \text{:-}\ L(x_1, x_2, x_3) \wedge L(x_4, 3, x_5) \wedge P(x_5, x_6) \\ ans(x_1, \text{'db'}) & \text{:-}\ L(x_1, 3, x_2) \wedge P(x_2, \text{'db'}). \end{array}$$

The goal of UCQ-optimization is usually twofold [7,24], namely to minimize the number of CQs and to minimize the number of atoms in each CQ. In the above UCQ, we would thus delete the second CQ and, moreover, eliminate the first atom from the body of the first CQ. In total, the above UCQ can be replaced by a single CQ $ans(x_4, x_6)$:- $L(x_4, 3, x_5) \wedge P(x_5, x_6)$. Analogously, we would naturally reduce the set Σ of two s-t tgds in Example 1 to the singleton $\Sigma' = \{L(x_4, 3, x_5) \wedge P(x_5, x_6) \rightarrow C(x_4, x_6)\}$.

As mentioned above, GAV mappings are only a special case of schema mappings given by s-t tgds which, in the general case, may have existentially quantified variables and conjunctions of atoms in the conclusion. Note that the existentially quantified variables are used to represent incomplete data (in the form of marked nulls [15]) in the target instance. Hence, as an additional optimization goal, we would like to minimize the number of existentially quantified variables in each s-t tgd. Moreover, we would now also like to minimize the number of atoms in the CQ of the conclusion.

Example 2 We revisit Example 1 and consider a new mapping $\mathcal{M}$ in the reverse direction so to speak: Let $\mathcal{M} = \langle \mathbf{S}, \mathbf{T}, \Sigma \rangle$ with $\mathbf{S} = \{C(\cdot,\cdot)\}$ and $\mathbf{T} = \{L(\cdot,\cdot,\cdot), P(\cdot,\cdot)\}$ where L, P, and C are as before. Moreover, let Σ be defined as follows:

$$\begin{aligned} \Sigma = \{C(x_1, x_2) &\rightarrow (\exists y_1, y_2, y_3, y_4) L(y_1, y_2, y_3) \wedge \\ &\qquad L(x_1, 3, y_4) \wedge P(y_4, x_2), \\ C(x_1, \text{'db'}) &\rightarrow (\exists y_1) L(x_1, 3, y_1) \wedge P(y_1, \text{'db'})\} \end{aligned}$$

Clearly, Σ is equivalent to the singleton

$$\Sigma' = \{C(x_1, x_2) \rightarrow (\exists y_4) L(x_1, 3, y_4) \wedge P(y_4, x_2)\}. \quad \square$$

The above schema mapping corresponds to the special case of LAV (local-as-view) [18] with s-t tgds of the form $A(\mathbf{x}) \rightarrow \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y})$, where the antecedent is a single atom $A(\mathbf{x})$ and all variables in $A(\mathbf{x})$ actually do occur in the conclusion. In the most general case (referred to as GLAV mappings), no restrictions are imposed on the CQs in the antecedent and conclusion nor on the variable occurrences. In order to formulate an optimality criterion for schema mappings with s-t tgds of this general form, the analogy with UCQs does not suffice. Indeed, the following example illustrates that we may get a highly unsatisfactory result if we just aim at the minimization of the number of s-t tgds and of the number of atoms inside each s-t tgd.

Example 3 Let $\mathcal{M} = \langle \mathbf{S}, \mathbf{T}, \Sigma \rangle$ with $\mathbf{S} = \{L(\cdot,\cdot,\cdot)\}$ and $\mathbf{T} = \{C(\cdot,\cdot), E(\cdot,\cdot)\}$ where L, and C are as before and E denotes the schema Equal-Year(course1, course2), i.e., E contains pairs of courses designed for students in the same year. Moreover, let Σ be defined as follows:

$$\begin{aligned}\Sigma = \{&L(x_1,x_2,x_3) \rightarrow (\exists y)C(x_1,y),\\ &L(x_1,x_2,x_3) \wedge L(x_4,x_2,x_5) \rightarrow E(x_1,x_4)\}\end{aligned}$$

Then Σ is equivalent to the singleton Σ' with the tgd

$$L(x_1,x_2,x_3) \wedge L(x_4,x_2,x_5) \rightarrow (\exists y)C(x_1,y) \wedge E(x_1,x_4)$$

Now suppose that the title-attribute is a key in Lecture. Let l_i denote the title of some lecture in a source instance I and suppose that I contains m lectures for students in the same year as l_i. Then the computation of the *canonical universal solution* (for details, see Section 2) yields two results of significantly different quality depending on whether we take Σ or Σ': In case of Σ, we get one tuple $C(l_i, y)$ with this course title l_i. In contrast, for Σ', we get m tuples $C(l_i, y_1), \ldots, C(l_i, y_m)$ with the same course title l_i. The reason for this is that the s-t tgd "fires" for every possible combination of key values x_1 and x_4, although for the conjunct $C(x_1, y)$ in the conclusion, only the value of x_1 is relevant. □

We shall refer to the two s-t tgds in Σ of the above example as the split form of the s-t tgd in Σ'. We shall formally define *splitting* of s-t tgds in Section 3. Intuitively, splitting aims at breaking up the conclusion of an s-t tgd in smaller parts such that the variables in the antecedent are indeed related to the atoms in the conclusion. Without this measure, any target instance would be artificially inflated with marked nulls as we have seen with Σ' in the above example. Splitting helps to avoid such anomalies. Indeed, it can be seen as an analogous operation to the decomposition of relational schemas into normal form where we also want to exclude that some attributes are fully determined by parts of a key. Carrying over this idea to s-t tgds, we want to exclude that some atoms in the conclusion are fully determined by parts of the atoms in the antecedent. Our first *optimization goal* for schema mappings will therefore be to minimize the number of s-t tgds only to the extent that splitting should be applied whenever possible. Minimizing the size of each s-t tgd and the number of existentially quantified variables in the conclusion will, of course, be pursued as another optimization goal. We thus have the following optimality criteria for sets Σ of s-t tgds:

- *cardinality-minimality*, i.e., the number of s-t tgds in Σ shall be minimal;
- *antecedent-minimality*, i.e., the total size of the antecedents of the s-t tgds in Σ shall be minimal;
- *conclusion-minimality*, i.e., the total size of the conclusions of the s-t tgds in Σ shall be minimal;
- *variable-minimality*, i.e., the total number of existentially quantified variables in the conclusions shall be minimal.

Then a set of s-t tgds is *optimal*, if it is minimal w.r.t. each of these four criteria. Following the above discussion, we only take s-t tgds into consideration for which no further splitting is possible. (We shall give a formal definition of this property and of the four optimality criteria in Section 3). Cardinality-minimality together with antecedent-minimality means that the *cost of the join-operations* is minimized when computing a canonical universal solution for some given source instance. Conclusion-minimality and variable-minimality mean that no unnecessary incomplete facts are introduced in the canonical universal solution. For the transformation of arbitrary sets of s-t tgds into optimal ones, we shall present a *novel system of rewrite rules*. Moreover, we shall show that the optimal form of a set of s-t tgds is *unique up to variable renaming*.

In other words, our optimization of schema mappings is also a *normalization of schema mappings*. As an immediate benefit of a normalization, we get a purely syntactical criterion for testing the equivalence of two schema mappings. Another, even more important application of such a normalization is in the area of defining the *semantics of query answering in data exchange*. Several definitions in this area depend on the concrete syntactic representation of the s-t tgds. This is, in particular, the case for queries with negated atoms (see e.g., [2,19]) and for aggregate queries (see [1]). This semantical dependence on the syntax of a mapping clearly is undesirable. Since the minimal set of s-t tgds produced by our rewrite rules is unique up to variable renaming, we can use it as the desired normal form which eliminates the effect of the concrete representation of the s-t tgds from the semantics of query answering.

Example 4 Consider a schema mapping $\mathcal{M} = \langle \mathbf{S}, \mathbf{T}, \Sigma \rangle$ with $\mathbf{S} = \{S(\cdot,\cdot,\cdot)\}$, $\mathbf{T} = \{L(\cdot,\cdot,\cdot), P(\cdot,\cdot)\}$, where L and P are as in Example 2. S denotes the relational schema Student(name, year, area). Moreover, let Σ express the following constraints: If there exists a student in any year, then there should exist at least one lecture for this year. Moreover, if a student specializes in a particular area, then there should be a professor in this area teaching at least one lecture for this year. We thus have the following set Σ with a single s-t tgd:

$$S(x_1, x_2, x_3) \to (\exists y_1, y_2, y_3, y_4, y_5)\ L(y_1, x_2, y_3) \land L(y_4, x_2, y_5) \land P(y_5, x_3)$$

Clearly, the first atom in the conclusion may be deleted. Now consider the source instance $I = \{S('\mathsf{bob}', 3, '\mathsf{db}')\}$ and suppose that we want to evaluate the query

$$ans(x_2) \coloneq L(x_1, x_2, x_3), \neg P(x_3, x_4)$$

over the target instance, i.e., we want to check if, in some year, there exists a lecture which has not been assigned to a professor. In [2,19], query answering via the "canonical universal solution" (for details, see Section 2) is proposed. Depending on whether the s-t tgd in Σ has been simplified or not, we either get $J = \{L(u_1, 3, u_2), L(u_3, 3, u_4), P(u_4, '\mathsf{db}')\}$ or the core thereof, $J' = \{L(u_1, 3, u_2), P(u_2, '\mathsf{db}')\}$ as the canonical universal solution. In the first case, the query yields the result $\{\langle 3 \rangle\}$ whereas, in the second case, we get $\emptyset$. □

Similarly, a unique normal form of the s-t tgds is crucial for the semantics of aggregate queries in data exchange, whose investigation has been initiated recently by Afrati and Kolaitis [1]. Aggregate queries are of the form SELECT f FROM R, where f is an aggregate operator min($R.A$), max($R.A$), count($R.A$), count($*$), sum($R.A$), or avg($R.A$), and where R is a target relation symbol or, more generally, a conjunctive query over the target schema and A is an attribute of R. On the one hand, [1] defines an interesting and non-trivial semantics of aggregate queries in data exchange. On the other hand, it is shown that the most important aggregate queries can be evaluated in polynomial time (data complexity). In this paper, we shall show how aggregate queries can benefit from our normalization of schema mappings.

So far, we have only mentioned mappings $\mathcal{M} = \langle \mathbf{S}, \mathbf{T}, \Sigma \rangle$, where Σ is a set of s-t tgds. In addition, Σ may contain constraints on the target instance alone. One of the most important forms of target constraints are equality-generating target-dependencies (egds, for short), which can be considered as a generalization of functional dependencies. Egds are formulas of the form $\forall \mathbf{x} (\varphi(\mathbf{x}) \to x_i = x_j)$ where φ is a CQ over $\mathbf{T}$ and x_i, x_j are variables in $\mathbf{x}$.

Example 5 We modify the setting from Example 1 and 2. Let $\mathcal{M} = \langle \mathbf{S}, \mathbf{T}, \Sigma \rangle$ with $\mathbf{S} = \{C(\cdot, \cdot, \cdot)\}$ and $\mathbf{T} = \{P(\cdot, \cdot, \cdot)\}$ where C and P denote the relational schemas Course (title, course-area, prof-area) and Prof(name, prof-area, course-area). The P-relation thus contains information on the area of the professor as well as on the area(s) of the courses taught by him/her. The set Σ of s-t tgds expresses the following constraints: For every course, there exists a professor who teaches courses in his/her own area of expertise and who teaches courses with this combination of course- and prof-area. Moreover, there exists a professor whose expertise matches the area of the course and vice versa. We thus define Σ as a mapping with the following two s-t tgds:

$$C(x_1, x_2, x_3) \to (\exists y_1, y_2)\ P(y_1, y_2, y_2) \land P(y_1, x_2, x_3)$$
$$C(x_1, x_2, x_3) \to (\exists y_1) P(y_1, x_3, x_2)$$

This set of dependencies is minimal. However, suppose that we add the egd $P(x_1, x_2, x_3) \to x_2 = x_3$, expressing that a professor only teaches courses in his/her own area of expertise. Then $P(y_1, y_2, y_2)$ can be eliminated from the conclusion of the first s-t tgd. Moreover, the first and the second s-t tgd imply each other. Hence, Σ can be replaced by either Σ' or Σ'' with

$$\Sigma' = \{C(x_1, x_2, x_3) \to (\exists y_1)\ P(y_1, x_2, x_3)\} \text{ and}$$
$$\Sigma'' = \{C(x_1, x_2, x_3) \to (\exists y_1)\ P(y_1, x_3, x_2)\}. \quad \square$$

Example 5 illustrates that, in the presence of target egds, our rewrite rules for the s-t tgds-only case are not powerful enough. To deal with target egds, we will introduce further rewrite rules. In particular, one of these new rewrite rules will result in the introduction of source egds to prevent situations where two sets of s-t tgds only differ on source instances which admit no target instance anyway. Indeed, in Example 5, Σ' and Σ'' only differ if $x_2 \neq x_3$ holds. But this is forbidden by the egd. Hence, Σ should be replaced by Σ^* with

$$\Sigma^* = \{C(x_1, x_2, x_3) \to x_2 = x_3, C(x_1, x_2, x_2) \to (\exists y_1)\ P(y_1, x_2, x_2)\}.$$

In summary, we shall be able to prove that our extended set of rewrite rules again leads to a *normal form* which is *unique up to variable renaming*. The main ingredients of our normalization and optimization are the splitting and simplification of tgds. In the presence of target egds, several pitfalls will have to be avoided when defining appropriate splitting and simplification rules so as not to destroy the uniqueness of the normal form.

Organization of the paper and summary of results. In Section 2, we recall some basic notions. A conclusion and an outlook to future work are given in Section 6. The main results of the paper are detailed in the Sections 3 – 5, namely:

- *Optimization and normalization of sets of s-t tgds.* In Section 3, we give a formal definition of the above mentioned optimality criteria for sets of s-t tgds and we present rewrite rules to transform any set of s-t tgds into an optimal one (i.e., minimal w.r.t. to these criteria). We shall also show that the normal form obtained by our rewrite rules is unique up to variable renaming. Moreover, we show that, if the length of each s-t tgd is bounded by a constant, then this normal form can be computed in polynomial time.

- *Extension to target egds.* In Section 4, the rewrite rule system for s-t tgds is then extended to schema mappings comprising target egds. Several non-trivial extensions (like the introduction of source egds) are required to arrive at a unique normal form again. The extended splitting and simplification rules will have to be defined very carefully so as not destroy this uniqueness.

- *Semantics of aggregate operators.* In Section 5, we discuss in detail the application of our normalization of schema mappings to the definition of a unique semantics of aggregate operators in data exchange.

2 Preliminaries

A *schema* $\mathbf{R} = \{R_1, \ldots, R_n\}$ is a set of relation symbols R_i each of a fixed arity. An *instance* over a schema $\mathbf{R}$ consists of a relation for each relation symbol in $\mathbf{R}$, s.t. both have the same arity. We only consider finite instances here.

Tuples of the relations may contain two types of *terms*: *constants* and *variables*. The latter are often also called *marked nulls* or *labeled nulls*. Two labeled nulls are equal iff they have the same label. For every instance J, we write $dom(J)$, $var(J)$, and $Const(J)$ to denote the set of terms, variables, and constants, respectively, of J. Clearly, $dom(J) = var(J) \cup Const(J)$ and $var(J) \cap Const(J) = \emptyset$. If we have no particular instance J in mind, we write *Const* to denote the set of all possible constants. We write $\mathbf{x}$ for a tuple $(x_1, x_2, \ldots, x_n)$. However, by slight abuse of notation, we also refer to the set $\{x_1, \ldots, x_n\}$ as $\mathbf{x}$. Hence, we may use expressions like $x_i \in \mathbf{x}$ or $\mathbf{x} \subseteq X$, etc.

Let $\mathbf{S} = \{S_1, \ldots, S_n\}$ and $\mathbf{T} = \{T_1, \ldots, T_m\}$ be schemas with no relation symbols in common. We call $\mathbf{S}$ the *source schema* and $\mathbf{T}$ the *target schema*. We write $\langle \mathbf{S}, \mathbf{T} \rangle$ to denote the schema $\{S_1, \ldots, S_n, T_1, \ldots, T_m\}$. Instances over $\mathbf{S}$ and $\mathbf{T}$ are called *source* and *target instances*, respectively. If I is a source instance and J a target instance, then their combination $\langle I, J \rangle$ is an instance of the schema $\langle \mathbf{S}, \mathbf{T} \rangle$.

Homomorphisms and substitutions. Let I, I' be instances. A *homomorphism* $h\colon I \to I'$ is a mapping $dom(I) \to dom(I')$, s.t. (1) whenever a fact $R(\mathbf{x}) \in I$, then $R(h(\mathbf{x})) \in I'$, and (2) for every constant c, $h(c) = c$. If such h exists, we write $I \to I'$. Moreover, if $I \leftrightarrow I'$ then we say that I and I' are *homomorphically equivalent*. In contrast, if $I \to I'$ but not vice versa, we say that I is *more general* than I', and I' is *more specific* than I.

If $h\colon I \to I'$ is invertible, s.t. h^{-1} is a homomorphism from I' to I, then h is called an *isomorphism*, denoted $I \cong I'$. An *endomorphism* is a homomorphism $I \to I$. An endomorphism is *proper* if it is not surjective (for finite instances, this is equivalent to being not injective), i.e., if it reduces the domain of I.

If I is an instance, and $I' \subseteq I$ is such that $I \to I'$ holds but for no other $I'' \subset I'\colon I \to I''$ (that is, I' cannot be further "shrunk" by a proper endomorphism), then I' is called a *core* of I. The core is unique up to isomorphism. Hence, we may speak about *the* core of I. Cores have the following important property: for arbitrary instances J and J', $J \leftrightarrow J'$ *iff* $core(J) \cong core(J')$.

A *substitution* σ is a mapping which sends variables to other domain elements (i.e., variables or constants). We write $\sigma = \{x_1 \leftarrow a_1, \ldots, x_n \leftarrow a_n\}$ if σ maps each x_i to a_i and σ is the identity outside $\{x_1, \ldots, x_n\}$. The application of a substitution is usually denoted in postfix notation, e.g., $x\sigma$ denotes the image of x under σ. For an expression $\varphi(\mathbf{x})$, which in the following will normally refer to a conjunctive query with variables in $\mathbf{x}$, we write $\varphi(\mathbf{x}\sigma)$ to denote the result of replacing every occurrence of every variable $x \in \mathbf{x}$ by $x\sigma$.

Schema Mappings and Data Exchange. A *schema mapping* is given by a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ where $\mathbf{S}$ is the source schema, $\mathbf{T}$ is the target schema, and Σ is a set of dependencies expressing the relationship between $\mathbf{S}$ and $\mathbf{T}$ and possibly also local constraints on $\mathbf{S}$ and $\mathbf{T}$. The *data exchange problem* associated with $\mathcal{M}$ is the following: Given a (ground) source instance I, find a target instance J, s.t. $\langle I, J \rangle \models \Sigma$. Such a J is called a *solution for* I or, simply, a *solution* if I is clear from the context. The set of all solutions for I under $\mathcal{M}$ is denoted by $Sol^{\mathcal{M}}(I)$. If $J \in Sol^{\mathcal{M}}(I)$ is such that $J \to J'$ holds for any other solution $J' \in Sol^{\mathcal{M}}(I)$, then J is called a *universal solution*. Since the universal solutions for a source instance I are homomorphically equivalent, the core of the universal solutions for I is unique up to isomorphism. It is the smallest universal solution [11].

In the following, we will often identify a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ with the set of dependencies Σ, without explicitly mentioning the schemas, for the sake of brevity.

Equivalence of schema mappings. Different notions of equivalence of schema mappings have been recently proposed by Fagin et al. [10]. In this paper, we will only consider the strongest one, namely logical equivalence.

Definition 1 [10] Two schema mappings Σ and Σ' over the schema $\langle \mathbf{S}, \mathbf{T} \rangle$ are logically equivalent (denoted as $\Sigma \equiv \Sigma'$) if, for every source instance I and target instance J, the equivalence $\langle I, J \rangle \models \Sigma \Leftrightarrow \langle I, J \rangle \models \Sigma'$ holds. In this case, the equality $Sol^{\Sigma}(I) = Sol^{\Sigma'}(I)$ holds for every source instance I.

Dependencies. *Embedded dependencies* [8] over a relational schema $\mathbf{R}$ are first-order formulas of the form

$$\forall \mathbf{x} (\varphi(\mathbf{x}) \to \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y}))$$

In case of *tuple-generating dependencies* (tgds), both *antecedent* φ and *conclusion* ψ are conjunctive queries (CQs) over the relation symbols from $\mathbf{R}$ such that all variables in $\mathbf{x}$ actually do occur in $\varphi(\mathbf{x})$. *Equality-generating dependencies* (egds) are of the form

$$\forall \mathbf{x} (\varphi(\mathbf{x}) \to x_i = x_j)$$

with $x_i, x_j \in \mathbf{x}$. Throughout this paper, we shall omit the universal quantifiers: By convention, all variables occurring in the antecedent are universally quantified (over the entire formula). In the context of data exchange, we are mainly dealing with *source-to-target dependencies* consisting of tuple-generating dependencies (or s-t tgds) over the schema $\langle \mathbf{S}, \mathbf{T} \rangle$ (the antecedent is a CQ over $\mathbf{S}$, the conclusion over $\mathbf{T}$) and *target dependencies* over $\mathbf{T}$. In the scope of this paper, target dependencies are restricted to equality-generating dependencies (referred to as "target egds"). Moreover, in Section 4, we shall also consider *source dependencies* consisting of egds over $\mathbf{S}$ (referred to as "source egds").

Database of a conjunctive query. Given a conjunctive query χ, we write $At(\chi)$ to denote the database comprising exactly the set of atoms of χ. If the variables of χ are instantiated with distinct fresh constants in $At(\chi)$, this database is called *frozen.* However, unless otherwise specified, we assume that $At(\chi)$ is not frozen, and that the variables of χ are instantiated with distinct labeled nulls in $At(\chi)$. If χ represents an antecedent or conclusion of some dependency τ, $At(\chi)$ is called the *antecedent* or, respectively, *conclusion database* of τ.

Chase. The data exchange problem can be solved by the *chase* [4], a sequence of steps, each enforcing a single constraint within some limited set of tuples. More precisely, let Σ contain a tgd $\tau: \varphi(\mathbf{x}) \to (\exists \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$, s.t. $I \models \varphi(\mathbf{a})$ for some assignment $\mathbf{a}$ on $\mathbf{x}$. Then we extend I with facts corresponding to $\psi(\mathbf{a}, \mathbf{z})$, where the elements of $\mathbf{z}$ are fresh labeled nulls. Note that this definition of the chase differs from the definition in [9], where no new facts are added if $I \vDash \exists \mathbf{y}\psi(\mathbf{a}, \mathbf{y})$ is already fulfilled. Omitting this check is referred to as *oblivious* [16] chase. It is the preferred version of chase if the result of the chase should not depend on the order in which the tgds are applied (see e.g., [2,19,1]).

Now suppose that Σ contains an egd $\varepsilon: \varphi(\mathbf{x}) \to x_i = x_j$, s.t. $I \models \varphi(\mathbf{a})$ for some assignment $\mathbf{a}$ on $\mathbf{x}$. This egd enforces the equality $a_i = a_j$. We thus choose a null a' among $\{a_i, a_j\}$ and replace *every occurrence* of a' in I by the other term; if $a_i, a_j \in Const(I)$ and $a_i \neq a_j$, the chase halts with *failure*. We write I^{Σ} to denote the result of chasing I with the dependencies Σ.

Consider an arbitrary schema mapping $\Sigma = \Sigma_{st} \cup \Sigma_t$ where Σ_{st} is a set of source-to-target tgds and Σ_t is a set of target egds. Then the solution to a source instance I can be computed as follows: We start off with the instance $\langle I, \emptyset \rangle$, i.e., the source instance is I and the target instance is initially empty. Chasing $\langle I, \emptyset \rangle$ with Σ_{st} yields the instance $\langle I, J \rangle$, where J is called the *preuniversal instance*. This chase always succeeds since Σ_{st} contains no egds. Then J is chased with Σ_t. This chase may fail on an attempt to unify distinct constants. If the chase succeeds, we end up with $U = J^{\Sigma_t}$, which is referred to as the *canonical universal solution* $CanSol^{\Sigma}(I)$ or, simply $CanSol(I)$. Both J and U can be computed in polynomial time w.r.t. the size of the source instance [9].

3 Normalization of s-t tgds

In this section, we investigate ways of optimizing sets of s-t tgds. In the first place, we thus formulate some natural optimality criteria. The following parameters of a set of s-t tgds will be needed in the definition of such criteria:

Definition 2 Let Υ be a set of s-t tgds. Then we define:

- $|\Upsilon|$ denotes the number of s-t tgds in Υ.
- $AntSize(\Upsilon) = \Sigma\{|At(\varphi(\mathbf{x}))| \colon \varphi(\mathbf{x}) \to \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y})$ is an s-t tgd in $\Upsilon\}$, i.e., $AntSize(\Upsilon)$ is the total number of atoms in all antecedents of tgds in Υ.
- $ConSize(\Upsilon) = \Sigma\{|At(\psi(\mathbf{x}, \mathbf{y}))| \colon \varphi(\mathbf{x}) \to \exists \mathbf{y}\, \psi(\mathbf{x}, \mathbf{y})$ is an s-t tgd in $\Upsilon\}$, i.e., $ConSize(\Upsilon)$ is the total number of atoms in all conclusions of tgds in Υ.
- $VarSize(\Upsilon) = \Sigma\{|\mathbf{y}| \colon \varphi(\mathbf{x}) \to \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y})$ is in $\Upsilon\}$, i.e., $VarSize(\Upsilon)$ is the total number of existentially quantified variables in all conclusions of tgds in Υ.

We would naturally like to transform any set of s-t tgds into an equivalent one where all the above parameters are minimal. Recall however our discussion on the splitting of s-t tgds from Example 3. As we pointed out there, the splitting of s-t tgds is comparable to normal form decomposition of relational schemas. It should clearly be applied in order to avoid anomalies like the introduction of obviously irrelevant atoms in the canonical universal solution as we saw in Example 3, where the set Σ (with two split s-t tgds) was certainly preferable to Σ' even though $|\Sigma'| < |\Sigma|$ and $AntSize(\Sigma') < AntSize(\Sigma)$ hold. Note that in Example 3, the equality $ConSize(\Sigma') = ConSize(\Sigma)$ holds. Intuitively, the effect of splitting is that the atoms in the conclusion of some s-t tgd are distributed over several strictly smaller s-t tgds. Thus, our goal should be to find an optimal set of s-t tgds (that is, a set where the above mentioned parameters are minimal) among those sets of s-t tgds for which no further splitting is possible. We now make precise what it means that "no further splitting" is possible and formally define optimality of a set of s-t tgds.

Definition 3 Let Σ be a set of s-t tgds. We say that Σ is *split-reduced* if there exists no Σ' equivalent to Σ, s.t. $|\Sigma'| > |\Sigma|$ but $ConSize(\Sigma') = ConSize(\Sigma)$.

Definition 4 Let Σ be a set of s-t tgds. We say that Σ is *optimal* if it is split-reduced, and if each of the parameters $|\Sigma|$, $AntSize(\Sigma)$, $ConSize(\Sigma)$, and $VarSize(\Sigma)$ is minimal among all split-reduced sets equivalent to Σ.

Of course, given an arbitrary set Σ of s-t tgds, it is a priori not clear that an optimal set Σ' equivalent to Σ exists, since it might well be the case that some Σ' minimizes some of the parameters while another set Σ'' minimizes the other parameters. The goal of this section is to show that optimality in the above sense can always be achieved and to construct an algorithm which transforms any set Σ of s-t tgds into an equivalent optimal one. To this end, we introduce a rewrite system which consists of two kinds of rewrite rules: rules which simplify each s-t tgd individually and rules which are applied to the entire set of s-t tgds. The following example illustrates several kinds of redundancy that a single s-t tgd may contain (and which may be eliminated with our rewrite rules).

Example 6 Consider the following dependency:

$$\begin{aligned}\tau\colon S(x_1,x_3) \wedge S(x_1,x_2) \rightarrow (\exists y_1,y_2,y_3,y_4,y_5)&\\ P(x_1,y_2,y_1) \wedge R(y_1,y_3,x_2) \wedge R(2,y_3,x_2)&\\ \wedge P(x_1,y_4,2) \wedge P(x_1,y_4,y_5) \wedge Q(y_4,x_3)&\end{aligned}$$

Clearly, τ is equivalent to the set $\{\tau_1,\tau_2\}$ of s-t tgds:

$$\begin{aligned}\tau_1\colon S(x_1,x_3) \wedge S(x_1,x_2) \rightarrow (\exists y_1,y_2,y_3)&\\ P(x_1,y_2,y_1) \wedge R(y_1,y_3,x_2) \wedge R(2,y_3,x_2)&\\ \tau_2\colon S(x_1,x_3) \wedge S(x_1,x_2) \rightarrow (\exists y_4,y_5)&\\ P(x_1,y_4,y_5) \wedge P(x_1,y_4,2) \wedge Q(y_4,x_3)&\end{aligned}$$

Now the antecedents of τ_1 and τ_2 can be simplified:

$$\begin{aligned}\tau_1'\colon S(x_1,x_2) \rightarrow (\exists y_1,y_2,y_3)&\\ P(x_1,y_2,y_1) \wedge R(y_1,y_3,x_2) \wedge R(2,y_3,x_2)&\\ \tau_2'\colon S(x_1,x_3) \rightarrow (\exists y_4,y_5)&\\ P(x_1,y_4,y_5) \wedge P(x_1,y_4,2) \wedge Q(y_4,x_3)&\end{aligned}$$

Finally, we may also simplify the conclusion of τ_2':

$$\tau_2''\colon S(x_1,x_3) \rightarrow (\exists y_4)P(x_1,y_4,2) \wedge Q(y_4,x_3)$$

In total, τ is equivalent to $\{\tau_1',\tau_2''\}$. □

For the simplifications illustrated in Example 6, we define the rewrite rules 1 – 3 in Figure 1. Rules 1 and 2 replace an s-t tgd τ by a simpler one (i.e., with fewer atoms) τ', while Rules 3 replaces τ by a set $\{\tau_1,\dots,\tau_n\}$ of simpler s-t tgds. These rules make use of the following definitions of the *core* and the *components* of CQs.

Definition 5 Let $\chi(\mathbf{u},\mathbf{v})$ be a CQ with variables in $\mathbf{u} \cup \mathbf{v}$ and let $\mathcal{A}$ denote the structure consisting of the atoms $At(\chi(\mathbf{u},\mathbf{v}))$, s.t. the variables $\mathbf{u}$ are considered as constants and the variables $\mathbf{v}$ as labeled nulls. Let $\mathcal{A}'$ denote the core of $\mathcal{A}$ with $\mathcal{A}' \subseteq \mathcal{A}$, i.e., there exists a substitution $\sigma\colon \mathbf{v} \rightarrow \mathit{Const} \cup \mathbf{u} \cup \mathbf{v}$ s.t. $At(\chi(\mathbf{u},\mathbf{v}\sigma)) = \mathcal{A}' \subseteq At(\chi(\mathbf{u},\mathbf{v}))$. Then we define the *core of* $\chi(\mathbf{u},\mathbf{v})$ as the CQ $\chi(\mathbf{u},\mathbf{v}\sigma)$.

Definition 6 Let $\chi(\mathbf{u},\mathbf{v})$ be a CQ with variables in $\mathbf{u} \cup \mathbf{v}$. We set up the *dual graph* $\mathcal{G}(\tau)$ as follows: The atoms of $\chi(\mathbf{u},\mathbf{v})$ are the vertices of $\mathcal{G}(\tau)$. Two vertices are connected if the corresponding atoms have at least one variable from $\mathbf{v}$ in common. Let $\{C_1,\dots,C_n\}$ denote the connected components of $\mathcal{G}(\tau)$. Moreover, for every $i \in \{1,\dots,n\}$, let $\mathbf{v}_i$ with $\emptyset \subseteq \mathbf{v}_i \subseteq \mathbf{v}$ denote those variables from $\mathbf{v}$, which actually occur in C_i and let $\chi_i(\mathbf{u},\mathbf{v}_i)$ denote the CQ consisting of the atoms in C_i. Then we define the *components of* $\chi(\mathbf{u},\mathbf{v})$ as the set $\{\chi_1(\mathbf{u},\mathbf{v}_1),\dots,\chi_n(\mathbf{u},\mathbf{v}_n)\}$.

Rewrite rules to simplify a set of s-t tgds

Rule 1 (Core of the conclusion, see Definition 5).
$\tau\colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x},\mathbf{y}) \implies$
$\tau'\colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x},\mathbf{y}\sigma)$,
s.t. $\psi(\mathbf{x},\mathbf{y}\sigma)$ is the core of $\psi(\mathbf{x},\mathbf{y})$.

Rule 2 (Core of the antecedent, see Definition 5).
$\tau\colon \varphi(\mathbf{x}_1,\mathbf{x}_2) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x}_1,\mathbf{y}) \implies$
$\tau'\colon \varphi(\mathbf{x}_1,\mathbf{x}_2\sigma) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x}_1,\mathbf{y})$,
s.t. $\varphi(\mathbf{x}_1,\mathbf{x}_2\sigma)$ is the core of $\varphi(\mathbf{x}_1,\mathbf{x}_2)$.

Rule 3 (Splitting, see Definition 6).
$\tau\colon\ \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x},\mathbf{y}) \implies \{\tau_1,\dots,\tau_n\}$, s.t.
$\{\psi_1(\mathbf{x},\mathbf{y}_1),\dots,\psi_n(\mathbf{x},\mathbf{y}_n)\}$ are the components of $\psi(\mathbf{x},\mathbf{y})$
and $\tau_i\colon\ \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y}_i)\psi_i(\mathbf{x},\mathbf{y}_i)$ for $i \in \{1,\dots,n\}$.

Rule 4 (Implication of an s-t tgd).
$\Sigma \implies \Sigma \setminus \{\tau\}$
if $\Sigma \setminus \{\tau\} \models \tau$.

Rule 5 (Implication of atoms in the conclusion).
$\Sigma \implies (\Sigma \setminus \{\tau\}) \cup \{\tau'\}$
if $\tau\colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x},\mathbf{y})$
and $\tau'\colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y}')\psi'(\mathbf{x},\mathbf{y}')$,
s.t. $At(\psi'(\mathbf{x},\mathbf{y}')) \subset At(\psi(\mathbf{x},\mathbf{y}))$
and $(\Sigma \setminus \{\tau\}) \cup \{\tau'\} \models \tau$.

Fig. 1 Redundancy elimination from a set of s-t tgds.

The splitting rule (i.e., Rule 3 in Figure 1) was already applied in Example 3. Rule 2 involving core computation of the antecedent was applied in Example 1, when we reduced $L(x_1,x_2,x_3) \wedge L(x_4,3,x_5) \wedge P(x_5,x_6)$ to its core $L(x_4,3,x_5) \wedge P(x_5,x_6)$. Likewise, in Example 6, the simplification of τ_1 and τ_2 to τ_1' and τ_2' is due to Rule 2. In a similar way, Rule 1 involving core computation of the conclusion allowed us to reduce $L(y_1,y_2,y_3) \wedge L(x_1,3,y_4) \wedge P(y_4,x_2)$ in Example 2 to $L(x_1,3,y_4) \wedge P(y_4,x_2)$. In Example 6, Rule 1 was applied when we replaced τ_2' by τ_2''.

The following example illustrates that additional rules are required in order to remove an s-t tgd or a part of an s-t tgd whose redundancy is due to the presence of other s-t tgds.

Example 7 Consider the set $\Sigma = \{\tau_1',\tau_2'',\tau_3\}$, where τ_1' and τ_2'' are the s-t tgds resulting from the simplification steps in Example 6 and τ_3 is given below:

$$\begin{aligned}\tau_1'\colon S(x_1,x_2) \rightarrow (\exists y_1,y_2,y_3)&\\ P(x_1,y_2,y_1) \wedge R(y_1,y_3,x_2) \wedge R(2,y_3,x_2)&\end{aligned}$$

$$\tau_2''\colon S(x_1,x_3) \rightarrow (\exists y_4)P(x_1,y_4,2) \wedge Q(y_4,x_3)$$

$$\tau_3\colon S(2,x) \rightarrow (\exists y)R(2,y,x)$$

The tgd τ_3 generates only a part of the atoms that τ_1' does, and fires in strictly fewer cases than τ_1'. Hence, τ_3 may be deleted. Moreover, considering the combined effect of the rules τ_1' and τ_2'', which fire on exactly the same tuples, and a substitution $\{y_1 \leftarrow 2, y_2 \leftarrow y_4\}$, we notice that the first two atoms in the conclusion of τ_1' are in fact redundant, and it is possible to reduce τ_1' to τ_1'': $S(x_1,x_2) \rightarrow (\exists y_3)R(2,y_3,x_2)$. In total, Σ may be replaced by $\Sigma' = \{\tau_1'',\tau_2''\}$. □

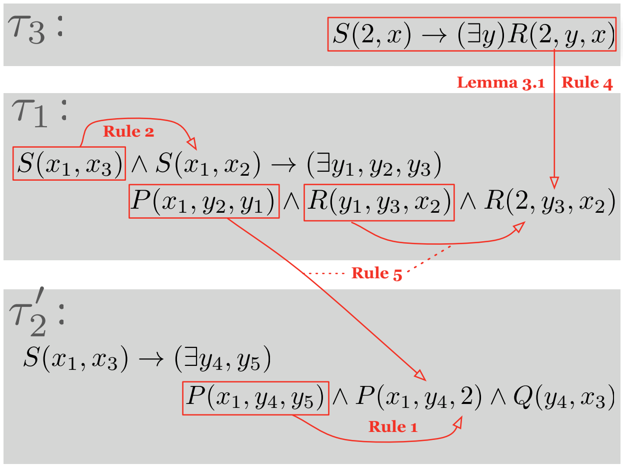
τ_3:
$S(2,x) \rightarrow (\exists y)R(2,y,x)$
Lemma 3.1
Rule 4
τ_1:
Rule 2
$S(x_1,x_3) \wedge S(x_1,x_2) \rightarrow (\exists y_1,y_2,y_3)$
$P(x_1,y_2,y_1) \wedge R(y_1,y_3,x_2) \wedge R(2,y_3,x_2)$
Rule 5
τ_2':
$S(x_1,x_3) \rightarrow (\exists y_4,y_5)$
$P(x_1,y_4,y_5) \wedge P(x_1,y_4,2) \wedge Q(y_4,x_3)$
Rule 1

Fig. 2 Tgd optimization. Rectangles mark eliminated atoms, arrows show justifications for elimination.

Rules 4 and 5 in Figure 1 allow us to eliminate such redundancies from a set Σ of s-t tgds: By Rule 4, we may delete an s-t tgd τ from Σ, if τ is implied by the others, like τ_3 in Example 7. Rule 5 allows us to replace a rule τ by a strictly smaller rule (with fewer atoms in the conclusion) if τ is implied by τ' together with the remaining s-t tgds in Σ (cf. the replacement of τ_1' with τ_1'' in Example 7 above). Figure 2 illustrates the elimination of redundant atoms via Rules 1, 2, 4 and 5 in a set $\{\tau_1, \tau_2', \tau_3\}$ of tgds from Examples 6 and 7.

In principle, the implication of a tgd by a set of dependencies can be tested by a procedural criterion based on the chase [4]. For our purposes, the following, declarative criterion is more convenient.

Lemma 1 *Consider an s-t tgd τ: $\varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x},\mathbf{y})$ and a set Σ of s-t tgds. Then $\Sigma \models \tau$ holds iff there exist (not necessarily distinct) s-t tgds $\tau_1, \ldots, \tau_k$ in Σ, such that all s-t tgds $\tau, \tau_1, \ldots, \tau_k$ are pairwise variable disjoint and the following conditions hold:*

(a) For every $i \in \{1, \ldots, k\}$, there exists a substitution λ_i: $\mathbf{x}_i \rightarrow Const \cup \mathbf{x}$, s.t. $At(\varphi_i(\mathbf{x}_i\lambda_i)) \subseteq At(\varphi(\mathbf{x}))$.

(b) A substitution μ: $\mathbf{y} \rightarrow Const \cup \mathbf{x} \cup \bigcup_{i=1}^{k} \mathbf{y}_i$ exists, s.t. $At(\psi(\mathbf{x},\mathbf{y}\mu)) \subseteq \bigcup_{i=1}^{k} At(\psi_i(\mathbf{x}_i\lambda_i,\mathbf{y}_i))$.

Proof For the "$\Rightarrow$"-direction, consider an arbitrary pair $\langle S,T\rangle$ of source and target instance, s.t. $\langle S,T\rangle \models \Sigma$. It is easy to show that, by conditions (a) and (b), then also $\langle S,T\rangle \models \tau$ holds. For the "$\Leftarrow$"-direction, we take the source instance $S = At(\varphi(\mathbf{x}))$, where we consider the variables $\mathbf{x}$ as constants. Moreover, let T denote the target instance which results from the *oblivious* chase of S with Σ. Let $\tau_1, \ldots, \tau_k$ denote the (not necessarily distinct) s-t tgds whose antecedent can be mapped into S via substitutions $\lambda_1, \ldots, \lambda_k$. These substitutions satisfy the condition (a). By $\langle S,T\rangle \models \Sigma$ and $\Sigma \models \tau$ we get the desired substitution μ for condition (b). □

Note that Rule 5 generalizes Rule 1 and, in principle, also Rule 4. Indeed, if we restrict Σ in Rule 5 to the singleton $\Sigma = \{\tau\}$, then the replacement of τ by τ' means that we reduce $\psi(\mathbf{x},\mathbf{y})$ to its core. Moreover, Rule 5 allows us to eliminate all atoms from the conclusion of τ iff τ may be deleted via Rule 4. Clearly, the deletion of the conclusion of τ essentially comes down to the deletion of τ itself. The correctness of Rules 1 – 5 is easily established (see Appendix A)

Lemma 2 *The Rules 1 – 5 in Figure 1 are correct, i.e.: Let Σ be a set of s-t tgds and $\tau \in \Sigma$. Suppose that Σ is transformed into Σ' by applying one of the Rules 1 – 5 to τ, that is:*

- *τ is replaced by a single s-t tgd τ' (via Rule 1,2,5),*
- *τ is replaced by s-t tgds $\tau_1, \ldots, \tau_n$ (via Rule 3),*
- *or τ is deleted (via Rule 4).*

Then Σ and Σ' are equivalent.

The following notion of a "proper instance" of an s-t tgd plays an important role for proving that our Rules 1 – 5 lead to a unique normal form. A proper instance of an s-t tgd τ is obtained from τ by eliminating at least one existentially quantified variable in the conclusion of τ, while keeping the antecedent unchanged.

Definition 7 Let τ: $\varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x},\mathbf{y})$ be an s-t tgd. We call an s-t tgd τ' a *proper instance* of τ, if there exists a strict subset $\mathbf{y}' \subset \mathbf{y}$ and a substitution σ: $\mathbf{y} \rightarrow Const \cup \mathbf{x} \cup \mathbf{y}'$, such that τ' is of the form τ': $\varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y}')\psi(\mathbf{x},\mathbf{y}\sigma)$.

Example 8 In the following three tgds, each next tgd is a proper instance of the previous ones:

τ_1: $S(x_1,x_2) \rightarrow (\exists y_1,y_2)Q(x_1,y_1,y_2)$

τ_2: $S(x_1,x_2) \rightarrow (\exists y_1)Q(x_1,y_1,y_1)$

τ_3: $S(x_1,x_2) \rightarrow Q(x_1,x_2,x_2)$

Moreover, observe that $\tau_2 \models \tau_1$ and $\tau_3 \models \tau_2$ holds. □

The importance of "proper instances" to our investigations comes from the following properties:

Lemma 3 *Let τ and τ' be s-t tgds, s.t. τ' is a proper instance of τ. Then the following properties hold:*

(1) $\tau' \models \tau$.

(2) Suppose that τ is reduced with respect to Rule 1. Then $\tau \not\models \tau'$.

Proof (Sketch) The proof of the first claim is easy.

Consider two tgds τ: $\varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\ \psi(\mathbf{x},\mathbf{y})$ and τ': $\varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y}')\,\psi(\mathbf{x},\mathbf{y}\sigma)$. Let $\langle S,T\rangle$ be an arbitrary pair of source and target instances with $\langle S,T\rangle \models \tau'$ and let λ: $\mathbf{x} \rightarrow dom(S)$ be a substitution, such that $At(\varphi(\mathbf{x}\,\lambda) \subseteq S$. We show that then also $T \models \psi(\mathbf{x}\lambda,\mathbf{y})$ holds. By $\langle S,T\rangle \models \tau'$, we get $T \models \psi(\mathbf{x}\lambda,\mathbf{y}\sigma)$, i.e., there exists a substitution μ, s.t. $At(\psi(\mathbf{x}\lambda,\mathbf{y}\sigma\mu)) \subseteq T$. But then, for $\nu = \sigma\mu$, we have $At(\psi(\mathbf{x},\mathbf{y}\sigma)) \subseteq T$. Thus, $T \models \psi(\mathbf{x}\lambda,\mathbf{y})$ holds.

For the second one, suppose that $\tau \models \tau'$ holds. We have to show that then Rule 1 is applicable to τ. Let $\langle S, T\rangle$ denote a pair of source and target instance with $S = At(\varphi(\mathbf{x}))$ and $T = At(\psi(\mathbf{x},\mathbf{y}))$. The variables in $\mathbf{x}$ are thus considered as constants while $\mathbf{y}$ are labeled nulls. Clearly, $\langle S,T\rangle \models \tau$ and $S \models \varphi(\mathbf{x})$. Thus, by the assumption $\tau \models \tau'$, also $T \models \psi(\mathbf{x},\mathbf{y}\sigma)$ holds, i.e., there exists a substitution $\mu\colon \mathbf{y}' \to dom(T)$ such that $At(\psi(\mathbf{x},\mathbf{y}\sigma\mu)) \subseteq T$. Hence, also the following inclusion holds: $At(\psi(\mathbf{x},\mathbf{y}\sigma\mu)) \subseteq At(\psi(\mathbf{x},\mathbf{y}))$. Note that $\mathbf{y}' = \mathbf{y}\sigma \subset \mathbf{y}$. Hence, also $At(\psi(\mathbf{x},\mathbf{y}\sigma\mu)) \subset At(\psi(\mathbf{x},\mathbf{y}))$. But then $At(\psi(\mathbf{x},\mathbf{y}))$ is not a core and, therefore, Rule 1 is applicable to τ. □

Lemma 4 *Let τ be an s-t tgd reduced w.r.t. the Rules 1 and 3 and let Σ be a set of s-t tgds. If $\Sigma \models \tau$, then one of the following two conditions is fulfilled: Either*

- *there exists a single s-t tgd $\tau_0 \in \Sigma$, s.t. $\tau_0 \models \tau$, or*
- *there exists a proper instance τ' of τ, s.t. $\Sigma \models \tau'$.*

Proof Let $\{\tau_1,\dots,\tau_k\} \subseteq \Sigma\setminus\{\tau\}$ with $\{\tau_1,\dots,\tau_k\} \models \tau$. Suppose that k is minimal with this property and that $k \geq 2$. We show that then $\{\tau_1,\dots,\tau_k\} \models \tau'$ holds for some proper instance τ' of τ. For $i \in \{1,\dots,k\}$, let τ_i's be pairwise variable disjoint and have the form $\tau_i\colon\ \varphi_i(\mathbf{x}_i) \to (\exists \mathbf{y}_i)\psi_i(\mathbf{x}_i,\mathbf{y}_i)$. By Lemma 1 and the definition of Rule 4, the τ_i's fulfill the following properties:

(a) For every $i \in \{1,\dots,k\}$, there exists a substitution $\lambda_i\colon \mathbf{x}_i \to \mathit{Const} \cup \mathbf{x}$, s.t. $At(\varphi_i(\mathbf{x}_i\lambda_i)) \subseteq At(\varphi(\mathbf{x}))$.

(b) A substitution $\mu\colon \mathbf{y} \to \mathit{Const} \cup \mathbf{x} \cup \bigcup_{i=1}^{k} \mathbf{y}_i$ exists s.t. $At(\psi(\mathbf{x},\mathbf{y}\mu)) \subseteq \bigcup_{i=1}^{k} At(\psi_i(\mathbf{x}_i\lambda_i,\mathbf{y}_i))$.

Let $At(\psi(\mathbf{x},\mathbf{y})) = \{A_1,\dots,A_n\}$. Clearly, $n \geq k \geq 2$. Suppose that $\{A_1\mu,\dots,A_\alpha\mu\} \subseteq At(\psi_1(\mathbf{x}_1\lambda_1,\mathbf{y}_1))$ while $\{A_{\alpha+1}\mu,\ \dots, A_n\mu\} \subseteq \bigcup_{i=2}^{k} At(\psi_i(\mathbf{x}_i\lambda_i,\mathbf{y}_i))$. By assumption, τ is reduced w.r.t. Rule 3, i.e., the conclusion of τ either consists of a single atom without variables from $\mathbf{y}$ or of atoms forming a single connected component of the dual graph $\mathcal{G}(\tau)$. By $n \geq k \geq 2$, the former case can be excluded. Hence, the atoms in $\{A_1,\dots,A_\alpha\}$ and $\{A_{\alpha+1},\dots,A_n\}$ share at least one variable $y \in \mathbf{y}$, i.e., y occurs in some atom A_i with $i \in \{1,\dots,\alpha\}$ and in some atom A_j with $j \in \{\alpha+1,\dots,n\}$. Let $\ell \neq 1$ denote the index, s.t. $A_j\mu \in At(\psi_\ell(\mathbf{x}_\ell\lambda_\ell,\mathbf{y}_\ell))$. In total, we thus have $A_i\mu \in At(\psi_1(\mathbf{x}_1\lambda_1,\mathbf{y}_1))$ and, therefore, $y\mu \in \mathit{Const} \cup \mathbf{x} \cup \mathbf{y}_1$. On the other hand, $A_j\mu \in At(\psi_\ell(\mathbf{x}_\ell\lambda_\ell,\mathbf{y}_\ell))$ and, therefore, $y\mu \in \mathit{Const} \cup \mathbf{x} \cup \mathbf{y}_\ell$. By assumption, $\mathbf{y}_1$ and $\mathbf{y}_\ell$ are disjoint. Thus, $y\mu \in \mathit{Const} \cup \mathbf{x}$.

We construct the desired proper instance τ' of τ as follows: Let $\mathbf{y}' := \mathbf{y} \setminus \{y\}$ and define the substitution $\sigma\colon \mathbf{y} \to \mathit{Const} \cup \mathbf{x} \cup \mathbf{y}'$, s.t. $y\sigma = y\mu$ and σ maps all other variables in $\mathbf{y}$ onto themselves. Then we have $\sigma\mu = \mu$, i.e., for every $y_i \in \mathbf{y}$, $y_i\sigma\mu = y_i\mu$. Clearly, $\{\tau_1,\dots,\tau_k\} \models \tau\mu$ and, therefore, also $\{\tau_1,\dots,\tau_k\} \models \tau\sigma$ holds. Hence, τ' is the desired proper instance of τ. □

Lemma 5 *Suppose that an s-t tgd $\tau \in \Sigma$ is reduced w.r.t. Rules 1 and 3 and that τ cannot be deleted via Rule 4. If there exists a proper instance τ' of τ, s.t. $\Sigma \models \tau'$ holds, then there exists an s-t tgd τ'', s.t. τ may be replaced by τ'' via Rule 5.*

Proof Let $\tau'\colon \varphi(\mathbf{x}) \to (\exists \mathbf{y}')\psi'(\mathbf{x},\mathbf{y}')$ and suppose that $\Sigma \models \tau'$ holds. Then there exist s-t tgds $\tau_1,\dots,\tau_k$ in Σ of the form $\tau_i\colon \varphi_i(\mathbf{x}_i) \to (\exists \mathbf{y}_i)\psi_i(\mathbf{x}_i,\mathbf{y}_i)$, s.t. the conditions (a) and (b) of Lemma 1 are fulfilled, i.e.:

(a) For every $i \in \{1,\dots,k\}$, there exists a substitution $\lambda_i\colon \mathbf{x}_i \to \mathit{Const} \cup \mathbf{x}$, s.t. $At(\varphi_i(\mathbf{x}_i\lambda_i)) \subseteq At(\varphi(\mathbf{x}))$.

(b) There exists a substitution $\mu\colon \mathbf{y} \to \mathit{Const} \cup \mathbf{x} \cup \bigcup_{i=1}^{k} \mathbf{y}_i$, s.t. $At(\psi'(\mathbf{x},\mathbf{y}'\mu)) \subseteq \bigcup_{i=1}^{k} At(\psi_i(\mathbf{x}_i\lambda_i,\mathbf{y}_i))$.

Let $\tau\colon \varphi(\mathbf{x}) \to (\exists \mathbf{y})\psi(\mathbf{x},\mathbf{y})$. We claim that at least one of the τ_i coincides with τ (up to variable renaming). Suppose to the contrary that $\tau_i \in \Sigma \setminus \{\tau\}$ holds for every $i \in \{1,\dots,k\}$. Then, the above conditions (a) and (b) imply that $\Sigma \setminus \{\tau\} \models \tau'$ holds by Lemma 1. Moreover, $\tau' \models \tau$ holds by Lemma 3, part (1). Thus, $\Sigma \setminus \{\tau\} \models \tau$ and τ could be deleted by Rule 4, which is a contradiction.

Let $I = \{i \mid 1 \leq i \leq k$, s.t. τ_i is obtained from τ via variable renaming$\}$. We define the CQ $\psi''(\mathbf{x},\mathbf{y}'')$ of the s-t tgd $\tau''\colon \varphi(\mathbf{x}) \to (\exists \mathbf{y}'')\psi''(\mathbf{x},\mathbf{y}'')$ as follows:

$\Theta = \{A(\mathbf{x},\mathbf{y}) \mid A(\mathbf{x},\mathbf{y}) \in At(\psi(\mathbf{x},\mathbf{y}))$ and $\exists i$, s.t. $i \in I$ and $A(\mathbf{x}\lambda_i,\mathbf{y}) \in At(\psi'(\mathbf{x},\mathbf{y}'\mu)) \cap At(\psi_i(\mathbf{x}_i\lambda_i,\mathbf{y}_i))\}$. Moreover, we set $\psi''(\mathbf{x},\mathbf{y}'') = \bigwedge_{A(\mathbf{x},\mathbf{y})\in\Theta} A(\mathbf{x},\mathbf{y})$.

Clearly, $(\Sigma \setminus \{\tau\}) \cup \{\tau''\} \models \tau'$ by Lemma 1. Thus, also $(\Sigma \setminus \{\tau\}) \cup \{\tau''\} \models \tau$, by Lemma 3, part (1). We claim that $At(\psi''(\mathbf{x},\mathbf{y}'')) \subset At(\psi(\mathbf{x},\mathbf{y}))$ holds. Suppose to the contrary that $At(\psi''(\mathbf{x},\mathbf{y}'')) = At(\psi(\mathbf{x},\mathbf{y}))$. Then, by the above definition of Θ and by Lemma 1, $\tau \models \tau'$ would hold. By Lemma 3, part (2), this implies that τ is not reduced w.r.t. Rule 1, which is a contradiction. Hence, τ'' is indeed the desired s-t tgd, s.t. τ may be replaced by τ'' via Rule 5. □

We now define a normal form of s-t tgds via the rewrite rules of this section. We will then show that this normal form is unique up to isomorphism in the sense defined below.

Definition 8 Let Σ be a set of s-t tgds and let Σ' be the result of applying the Rules 1 – 5 of Figure 1 exhaustively to Σ. Then Σ' is the *normal form* of Σ.

Definition 9 Let τ_1: $\varphi_1(\mathbf{x}_1) \to (\exists \mathbf{y}_1)\psi_1(\mathbf{x}_1,\mathbf{y}_1)$ and τ_2: $\varphi_2(\mathbf{x}_2) \to (\exists \mathbf{y}_2)\psi_2(\mathbf{x}_2,\mathbf{y}_2)$ be two tgds. We say that τ_1 and τ_2 are *isomorphic* if τ_2 is obtained from τ_1 via variable renamings $\eta\colon \mathbf{x}_1 \to \mathbf{x}_2$ and $\vartheta\colon \mathbf{y}_1 \to \mathbf{y}_2$.

Let Σ_1 and Σ_2 be two sets of tgds. We say that Σ_1 and Σ_2 are *isomorphic* if $|\Sigma_1| = |\Sigma_2|$, every $\tau_1 \in \Sigma_1$ is isomorphic to precisely one $\tau_2 \in \Sigma_2$ and every $\tau_2 \in \Sigma_2$ is isomorphic to precisely one $\tau_1 \in \Sigma_1$.

We start by showing for two single s-t tgds τ_1 and τ_2 that logical equivalence and isomorphism coincide, provided that the s-t tgds are reduced via our rewrite rules. This result will then be extended to sets Σ_1 and Σ_2 of s-t tgds.

Lemma 6 *Let τ_1 and τ_2 be two s-t tgds and suppose that τ_1 and τ_2 are reduced w.r.t. Rules 1 – 3. Then τ_1 and τ_2 are isomorphic, iff τ_1 and τ_2 are equivalent.*

Proof (Sketch) The "$\Rightarrow$"-direction follows immediately from Lemma 1. For the "$\Leftarrow$"-direction, let τ_1 and τ_2 be equivalent s-t tgds with $\tau_1\colon \varphi_1(\mathbf{x}_1, \mathbf{x}_2) \to (\exists \mathbf{y})\psi_1(\mathbf{x}_1, \mathbf{y})$ and $\tau_2\colon \varphi_2(\mathbf{u}_1, \mathbf{u}_2) \to (\exists \mathbf{v})\psi(\mathbf{u}_1, \mathbf{v})$.

By Lemma 1, there exist substitutions λ and ρ, s.t.
$\lambda\colon \mathbf{x}_1 \cup \mathbf{x}_2 \to \mathit{Const} \cup \mathbf{u}_1 \cup \mathbf{u}_2$, and
$\rho\colon \mathbf{u}_1 \cup \mathbf{u}_2 \to \mathit{Const} \cup \mathbf{x}_1 \cup \mathbf{x}_2$, such that
$At(\varphi_1(\mathbf{x}_1\lambda, \mathbf{x}_2\lambda)) \subseteq At(\varphi_2(\mathbf{u}_1, \mathbf{u}_2))$ and
$At(\varphi_2(\mathbf{u}_1\rho, \mathbf{u}_2\rho)) \subseteq At(\varphi_1(\mathbf{x}_1, \mathbf{x}_2))$.

By exploiting the equivalence of τ_1 and τ_2 and the fact that these s-t tgds are reduced w.r.t. Rule 2, we can show that the antecedents of τ_1 and τ_2 are isomorphic (i.e., the above inclusions are in fact equalities). Moreover, by exploiting that the s-t tgds are reduced w.r.t. Rule 1, we may conclude that τ_1 and τ_2 are isomorphic. For details, see Appendix B. □

Theorem 1 *Let Σ_1 and Σ_2 be equivalent sets of s-t tgds, i.e., $\Sigma_1 \models \Sigma_2$ and $\Sigma_2 \models \Sigma_1$. Let Σ'_1 and Σ'_2 denote the normal form of Σ_1 and Σ_2, respectively. Then Σ'_1 and Σ'_2 are isomorphic.*

Proof Let Σ_1 and Σ_2 be equivalent. Moreover, let Σ'_1 and Σ'_2 denote the normal form of Σ_1 and Σ_2, respectively. By the correctness of our rewrite rules 1 – 5, of course, also Σ'_1 and Σ'_2 are equivalent.

We first show that every s-t tgd in Σ'_1 is isomorphic to some s-t tgd in Σ'_2 and vice versa. Suppose to the contrary that this is not the case. W.l.o.g., we assume that there exists a $\tau \in \Sigma'_1$ which is not isomorphic to any s-t tgd in Σ'_2. By the equivalence of Σ'_1 and Σ'_2, the implication $\Sigma'_2 \models \tau$ clearly holds. By Lemma 4, either $\tau_0 \models \tau$ for a single s-t tgd $\tau_0 \in \Sigma'_2$ or there exists a proper instance τ' of τ, s.t. $\Sigma'_2 \models \tau'$.

We start by considering the case that $\tau_0 \models \tau$ for a single s-t tgd $\tau_0 \in \Sigma'_2$. By the equivalence of Σ'_1 and Σ'_2, the implication $\Sigma'_1 \models \tau_0$ holds and we can again apply Lemma 4, i.e., either $\tau_1 \models \tau_0$ for a single s-t tgd $\tau_1 \in \Sigma'_1$ or there exists a proper instance τ'_0 of τ_0, s.t. $\Sigma'_1 \models \tau'_0$. Again we consider first the case that a single s-t tgd is responsible for the implication. Actually, if τ_1 were identical to τ then we had the equivalence $\tau_1 \models \tau$ and $\tau \models \tau_1$. Since both τ and τ_1 are reduced w.r.t. Rules 1 – 3, this would mean (by Lemma 6) that τ_1 and τ are isomorphic. This contradicts our original assumption that τ is not isomorphic to any s-t tgd in Σ'_2. Hence, the case that $\tau_1 \models \tau_0$ for a single s-t tgd $\tau_1 \in \Sigma'_1$ means that τ_1 is different from τ. In total, we thus have $\tau_1 \models \tau_0$ and $\tau_0 \models \tau$ and, therefore, $\tau_1 \models \tau$ for a s-t tgd $\tau_1 \in \Sigma'_1 \setminus \{\tau\}$. Hence, τ can be deleted from Σ'_1 via Rule 4, which contradicts the normal form of Σ'_1.

It thus remains to consider the cases that there exists a proper instance τ' of τ, s.t. $\Sigma'_2 \models \tau'$ or there exists a proper instance τ'_0 of τ_0, s.t. $\Sigma'_1 \models \tau'_0$. We only show that the first one leads to a contradiction. The second case is symmetric. So suppose that there exists a proper instance τ' of τ, s.t. $\Sigma'_2 \models \tau'$. By the equivalence of Σ'_1 and Σ'_2, we have $\Sigma'_1 \models \Sigma'_2$ and, therefore, also $\Sigma'_1 \models \tau'$. But then τ can be replaced in Σ'_1 by τ' via Rule 5. Hence, by Lemma 5, τ can be replaced in Σ'_1 by some s-t tgd τ'' via Rule 5. But this contradicts the assumption that Σ'_1 is in normal form.

Hence, it is indeed the case that every s-t tgd in Σ'_1 is isomorphic to *some* s-t tgd in Σ'_2 and vice versa. We claim that every s-t tgd in Σ'_1 is isomorphic to *precisely one* s-t tgd in Σ'_2 and vice versa. Suppose to the contrary that there exists a s-t tgd τ which is isomorphic to two s-t tgds τ_1 and τ_2 in the other set. W.l.o.g., $\tau \in \Sigma'_1$ and $\tau_1, \tau_2 \in \Sigma'_2$. Clearly, τ_1 and τ_2 are isomorphic since they are both isomorphic to τ. Hence, $\tau_1 \models \tau_2$ and, therefore, $\Sigma'_2 \setminus \{\tau_2\} \models \tau_2$, i.e., Rule 4 is applicable to Σ'_2, which contradicts the assumption that Σ'_2 is in normal form. □

We now consider the complexity of computing the normal form of a set of s-t tgds. Of course, the application of any of the Rules 1, 2, 4, and 5 is NP-hard, since they involve CQ answering. However, below we show that if the length of each s-t tgd (i.e., the number of atoms) is bounded by a constant, then the normal form according to Definition 8 can be obtained in polynomial time.

Note that a constant upper bound on the length of the s-t tgds is a common restriction in data exchange since, otherwise, even the most basic tasks like, computing a target instance fulfilling all s-t tgds, would be intractable.

Theorem 2 *Suppose that the length (i.e., the number of atoms) of the s-t tgds under consideration is bounded by some constant b. Then there exists an algorithm which reduces an arbitrary set Σ of s-t tgds to normal form in polynomial time w.r.t. the total size $||\Sigma||$ of (an appropriate representation of) Σ.*

Proof (Sketch) First, the total number of applications of each rule is bounded by the total number of atoms in all s-t tgds in Σ. Indeed, Rule 4 deletes an s-t tgd. The Rules 1, 2, and 5 delete at least one atom from an s-t tgd. Rule 3 splits the conclusion of an s-t tgd in 2 or more parts. Hence, also the total number of applications of Rule 3 is bounded by the total number of atoms in

Σ. Finally, the application of each rule is feasible in polynomial time since the most expensive part of these rules is the CQ answering where the length of the CQs is bounded by the number of atoms in a single s-t tgd.

For details, see Appendix C. □

The restriction on the number of atoms in each s-t tgd is used in the above proof only in order to show that each rule application is feasible in polynomial time. The argument that the total number of rule applications is bounded by the total number of atoms in all s-t tgds in Σ applies to any set Σ of s-t tgds. We thus get:

Corollary 1 *The rewrite rule system consisting of Rules 1 – 5 is terminating, i.e., Given an arbitrary set Σ of s-t tgds, the non-deterministic, exhaustive application of the Rules 1 – 5 terminates.*

It can be shown that the unique normal form produced by our rewrite rules is indeed *optimal*.

Theorem 3 *Let Σ be a set of s-t tgds in normal form. Then Σ is optimal according to Definition 4.*

Proof The proof proceeds in three stages:

(1) Σ is split-reduced. Suppose to the contrary that it is not. Then there exists a set Σ' with $\Sigma \equiv \Sigma'$, $|\Sigma| < |\Sigma'|$ and $\mathit{ConSize}(\Sigma) = \mathit{ConSize}(\Sigma')$. Let Σ^* denote the result of exhaustively applying our rewrite rules to Σ'. By Theorem 1, Σ and Σ^* are isomorphic. Hence, we have $\mathit{ConSize}(\Sigma) = \mathit{ConSize}(\Sigma^*)$ and $|\Sigma| = |\Sigma^*|$. An inspection of the Rules 1 – 5 reveals that they may possibly decrement the value of *ConSize*() (by either deleting an atom in the conclusion or deleting an entire s-t tgd) but they never increment the value of *ConSize*(). By $\mathit{ConSize}(\Sigma) = \mathit{ConSize}(\Sigma')$ together with $\mathit{ConSize}(\Sigma) = \mathit{ConSize}(\Sigma^*)$, we immediately have $\mathit{ConSize}(\Sigma') = \mathit{ConSize}(\Sigma^*)$. Hence, when transforming Σ' into Σ^*, we never decrement *ConSize*() and, thus, we never delete an s-t tgd. But then $|\Sigma'| = |\Sigma^*|$ and, therefore, $|\Sigma'| = |\Sigma|$, which contradicts the assumption that $|\Sigma| < |\Sigma'|$ holds.

(2) $\mathit{ConSize}(\Sigma)$ and $\mathit{VarSize}(\Sigma)$ are minimal. It is easy to verify that by no application of any of the Rules 1 – 5 the parameters *ConSize*() or *VarSize*() can increase, i.e., if a set Υ of s-t tgds is obtained from some set Υ' by an application of one of the Rules 1 – 5, then $\mathit{ConSize}(\Upsilon) \leq \mathit{ConSize}(\Upsilon')$ and $\mathit{VarSize}(\Upsilon) \leq \mathit{VarSize}(\Upsilon')$.

Now let Σ' be a set of s-t tgds equivalent to Σ, and let Σ^* denote the result of exhaustively applying our rewrite rules to Σ'. By Theorem 1, Σ and Σ^* are isomorphic. Hence, we have the following relations: $\mathit{ConSize}(\Sigma) = \mathit{ConSize}(\Sigma^*) \leq \mathit{ConSize}(\Sigma')$ and also $\mathit{VarSize}(\Sigma) = \mathit{VarSize}(\Sigma^*) \leq \mathit{VarSize}(\Sigma')$.

(3) $|\Sigma|$ and $\mathit{AntSize}(\Sigma)$ are minimal. Let Σ' be an arbitrary split-reduced set of s-t tgds equivalent to Σ. We first show that $|\Sigma| \leq |\Sigma'|$. Suppose to the contrary that $|\Sigma| > |\Sigma'|$. We derive a contradiction by showing that then Σ' is not split-reduced. By (2), we know that $\mathit{ConSize}(\Sigma) \leq \mathit{ConSize}(\Sigma')$ holds. Analogously to the proof of Lemma 7, we can transform Σ into $\bar{\Sigma}$ with $\mathit{ConSize}(\bar{\Sigma}) = \mathit{ConSize}(\Sigma')$ simply by choosing an s-t tgd τ in Σ and inflating its conclusion by sufficiently many atoms of the form $P(u_1, \ldots, u_k)$. In total, we then have $\bar{\Sigma} \equiv \Sigma'$, $\mathit{ConSize}(\bar{\Sigma}) = \mathit{ConSize}(\Sigma')$, and $|\bar{\Sigma}| > |\Sigma'|$. Hence, Σ' is not split-reduced.

It remains to prove $\mathit{AntSize}(\Sigma) \leq \mathit{AntSize}(\Sigma')$ as the final inequality. It is easy to verify that the parameter *AntSize*() can never increase when one of the Rules 1, 2, 4, or 5 is applied. Moreover, by Lemma 7, we know that Rule 3 is never applicable when we transform a split-reduced set of s-t tgds into normal form. Now let Σ^* denote the normal form of Σ'. By Theorem 1, Σ and Σ^* are isomorphic. Hence, we have $\mathit{AntSize}(\Sigma) = \mathit{AntSize}(\Sigma^*) \leq \mathit{AntSize}(\Sigma')$. □

An important motivation for seeking a conclusion-minimal mapping Σ is to keep the redundancies in the target instance small when using Σ in data exchange. The following theorem establishes that our normal form indeed serves this purpose.

Theorem 4 *Let $\mathcal{M} = \langle \mathbf{S}, \mathbf{T}, \Sigma \rangle$ be a schema mapping where Σ is a set of s-t tgds and Σ is in normal form. Moreover, let Σ' be another set of s-t tgds, s.t. Σ and Σ' are equivalent and let I be an arbitrary source instance. Then there exists a variable renaming λ on the variables in the canonical universal solution $\mathit{CanSol}^{\Sigma}(I)$, s.t. $\mathit{CanSol}^{\Sigma}(I)\lambda \subseteq \mathit{CanSol}^{\Sigma'}(I)$ holds, i.e., the canonical instance produced by Σ is subset-minimal up to variable renaming.*

Proof It is easy to verify that every application of any of the Rules 1 – 5 either leaves the corresponding canonical universal solution unchanged or prevents the introduction of some atoms in the canonical universal solution, i.e., let the set Υ of s-t tgds be obtained from some set Υ' by an application of one of the Rules 1 – 5, then there exists a substitution μ, s.t. $\mathit{CanSol}^{\Upsilon}(I)\mu \subseteq \mathit{CanSol}^{\Upsilon'}(I)$ holds. The theorem follows by an easy induction argument. □

We conclude this section by two remarks on the splitting rule:

(1) The purpose of the splitting rule is to enable a further simplification of the antecedents of the resulting s-t tgds. Of course, it may happen that no further simplification is possible. As an example, consider a schema mapping $\Sigma = \{R(x,y) \wedge R(y,z) \rightarrow S(x,z) \wedge T(z,x)\}$. Splitting yields $\Sigma' = \{R(x,y) \wedge R(y,z) \rightarrow S(x,z);\ R(x,y) \wedge R(y,z) \rightarrow T(z,x)\}$, which cannot be further simplified. In cases like this, one may either "undo" the splitting or simply keep track of s-t tgds with identical

(possibly up to variable renaming) antecedents in order to avoid multiple evaluation of the same antecedent by the chase.

(2) Definition 3 gives a "semantical" definition of "split reduced" while the splitting rule is a "syntactical" criterion. The following lemma establishes the close connection between them.

Lemma 7 *Let Σ be a split-reduced set of s-t tgds and let Σ^* denote the normal form of Σ. Then, for every possible sequence of rewrite rule applications, this normal form Σ^* is obtained from Σ without ever applying Rule 3 (i.e., splitting).*

Proof Suppose to the contrary that there exists a sequence of rewrite rule applications including the splitting rule on the way from Σ to Σ^*. An inspection of the rewrite rules shows that an application of Rule 4 (i.e., deletion of an s-t tgd) is never required as a precondition in order to be able to apply another rule. Hence, w.l.o.g., we may assume that Rule 4 is applied at the very end of the transformation of Σ into Σ^*, so that Rule 4 does not precede the application of any other rule. Let $\Sigma_0, \dots, \Sigma_n$ with $\Sigma_0 = \Sigma$ and $\Sigma_n = \Sigma^*$ denote the sequence of intermediate results along this transformation of Σ into Σ^*. Then there exists an $i \in \{1, \dots, n\}$, s.t. Σ_i is obtained from Σ_{i-1} by an application of Rule 3. Moreover, suppose that this is the first application of Rule 3 along this transformation of Σ into Σ^*. Since we are assuming that all applications of Rule 4 occur at the very end of this transformation from Σ to Σ^*, we have $|\Sigma_{i-1}| = |\Sigma|$ and, therefore, $|\Sigma_i| > |\Sigma|$. An inspection of the Rules 1, 2, 3, and 5 reveals that they may possibly decrement the value of *ConSize*() (by deleting an atom in the conclusion via Rule 1 or 5) but they never increment the value of *ConSize*(). Hence, we have $\mathit{ConSize}(\Sigma_i) \leq \mathit{ConSize}(\Sigma)$. We derive a contradiction by constructing a set Σ' equivalent to Σ_i (and, hence, to Σ), with $\mathit{ConSize}(\Sigma') = \mathit{ConSize}(\Sigma)$ and $|\Sigma'| > |\Sigma|$. In other words, we show that Σ is not split-reduced.

Let τ with $\tau\colon \varphi(\mathbf{x}) \to \exists \mathbf{y}\ \psi(\mathbf{x}, \mathbf{y})$ be an arbitrary s-t tgd in Σ_i and let $P(z_1, \dots, z_k)$ with $\{z_1, \dots, z_k\} \subseteq \mathbf{x} \cup \mathbf{y}$ be an atom in the conclusion of τ. Clearly, we may add atoms of the form $P(u_1, \dots, u_k)$ for fresh, existentially quantified variables $u_1, \dots, u_k$ to the conclusion without changing the semantics of Σ. Indeed, any such atom could be removed again by our Rule 1 (Core of the conclusion). Then we transform Σ_i into Σ' with $\mathit{ConSize}(\Sigma') = \mathit{ConSize}(\Sigma_i)$ simply by inflating the conclusion of τ in Σ_i by sufficiently many atoms of the form $P(u_1, \dots, u_k)$. In total, we then have $\Sigma' \equiv \Sigma_i \equiv \Sigma$, $\mathit{ConSize}(\Sigma') = \mathit{ConSize}(\Sigma)$, and $|\Sigma'| = |\Sigma_i| > |\Sigma'|$. Hence, Σ is not split-reduced, which contradicts the assumption of this lemma. □

If a mapping Σ contains redundancies in the sense that one of the Rules 1, 4, 5 is applicable, then the notion of "split-reduced" according to Definition 3 and the non-applicability do not necessarily coincide as the following example illustrates. However, if Rules 1, 4, 5 are not applicable, then Definition 3 is exactly captured by the splitting rule (Rule 3).

Example 9 Consider the set $\Sigma = \{\tau\}$ of s-t tgds with $\tau\colon P(x_1, x_2) \to (\exists y_1, y_2) R(x_1, x_2, y_1) \wedge R(x_1, y_1, y_2)$. On the one hand, Rule 3 is not applicable because the conclusion of τ consists of a single connected component.

On the other hand, τ may be also simplified (via Rule 1 or Rule 5) to $\tau'\colon P(x_1, x_2) \to (\exists y) R(x_1, x_2, y)$. Now let Σ' consist of two "copies" of τ', i.e., $\Sigma' = \{\tau', \tau''\}$ with $\tau''\colon P(z_1, z_2) \to (\exists y) R(z_1, z_2, y)$. Then we have the equivalence $\Sigma \equiv \Sigma'$. Moreover, $|\Sigma'| > |\Sigma|$ and $\mathit{ConSize}(\Sigma') = \mathit{ConSize}(\Sigma)$. Hence, Σ is not split-reduced in the sense of Definition 3. □

Lemma 8 *Let Σ be a set of s-t tgds, s.t. Σ is reduced w.r.t. Rules 1, 4, 5. Then the following equivalence holds: Σ is split-reduced (according to Definition 3) iff Rule 3 (i.e., splitting) is not applicable.*

Proof If Rule 3 is applicable to Σ, then Σ can obviously be transformed into an equivalent set Σ' with $|\Sigma'| > |\Sigma|$ and $\mathit{ConSize}(\Sigma') = \mathit{ConSize}(\Sigma)$, i.e., Σ is not split-reduced according to Definition 3.

Now suppose that the splitting rule is not applicable to Σ. We have to show that then Σ is split-reduced. Suppose to the contrary that it is not split-reduced, i.e., there exists an equivalent set Σ' with $|\Sigma'| > |\Sigma|$ and $\mathit{ConSize}(\Sigma') = \mathit{ConSize}(\Sigma)$. We derive a contradiction by exploiting Theorem 1 (i.e., the uniqueness of the normal form according to Definition 8).

First, we observe that the normal form Σ^* of Σ can be obtained via Rule 2 only. Indeed, by assumption, none of Rules 1, 3, 4, 5 is applicable to Σ. Hence, either Σ already is in normal form (i.e., Rule 2 is not applicable either) or Σ can be simplified via Rule 2. Clearly, an application of Rule 2 does not enable the application of any of the other rules. Hence, Σ^* is obtained by iterated applications of Rule 2 only. Note that Rule 2 has no influence on the cardinality and on the conclusion-size of a mapping. Hence, we have $|\Sigma| = |\Sigma^*|$ and $\mathit{ConSize}(\Sigma) = \mathit{ConSize}(\Sigma^*)$.

Second, let us transform Σ' into normal form. By Theorem 1, this normal form is unique up to isomorphism. Hence, w.l.o.g., this normal form of Σ' is Σ^*. As far as the cardinality of the involved mappings is concerned, we have $|\Sigma'| > |\Sigma|$ and $|\Sigma| = |\Sigma^*|$. Hence, during the transformation of Σ' into $|\Sigma^*|$, eventually Rule 4 or 5 must be applied thus reducing the conclusion-size. Hence, we have $\mathit{ConSize}(\Sigma') > \mathit{ConSize}(\Sigma^*)$. But this is a contradiction to the above equalities $\mathit{ConSize}(\Sigma') = \mathit{ConSize}(\Sigma)$ and $\mathit{ConSize}(\Sigma) = \mathit{ConSize}(\Sigma^*)$ □

4 Extension to Target Egds

We now extend our rewrite rule system to schema mappings with both s-t tgds and target egds. Several additional considerations and measures are required to arrive at a unique normal form and a basis for the s-t tgd optimization also in this case. The outline of this section is as follows:

(1) We have already seen in Example 5 that the presence of egds may have an effect on the equivalence between two sets of s-t tgds. We shall therefore first present a method of "propagating" the effects of the egds into the s-t tgds.

(2) Splitting has played an important role in all our considerations so far. It will turn out that splitting via Rule 3 as in the tgd-only case is not powerful enough if egds are present. We shall therefore present a generalization of the notion of "split-reduced" and of the splitting rule to the case when also egds are present. This will lead to the notion of "egd-split-reduced" mappings.

(3) The intuition of "egd-split-reduced" mappings is that it is not possible to generate the atoms in the conclusion of some tgd by means of several tgds. The antecedent may thus possibly be left unchanged. It can easily be shown that, in general, there does not exist a unique "egd-split-reduced" normal form. We therefore restrict this notion to "antecedent-split-reduced" mappings, i.e.: a tgd is replaced by new tgds only if the new tgds have strictly smaller antecedent than the original one. With this concept, we shall manage to prove that there always exists a unique (up to isomorphism) normal form also in the presence of egds.

(4) Finally, we leave aside the considerations on splitting and concentrate on the optimization of the set of s-t tgds according to the criteria of Section 3. We shall show that grouping the s-t tgds by homomorphically equivalent antecedents is the key to any optimization tasks in this area.

(5) We also look at the operation opposite to splitting: namely, merge of s-t tgds with homomorphically equivalent antecedents. As we will show, unlike the s-t tgds only case, in presence of target dependencies the splitting of s-t tgds can cause an increase in the total number of conclusion atoms. Hence, for some cases the merge opeartion can be a reasonable alternative. We will show, however, that with respect to the unique normal form, the "merged" form of the mappings is no more useful than the "egd-split-reduced" form.

Propagating the effect of egds into s-t tgds. An important complication introduced by the egds has already been hinted at in Section 1, namely the equivalence of two sets of s-t tgds may be affected by the presence of egds:

Example 10 (Example 5 slightly extended).

$$\begin{aligned}\Sigma_{st} = \{&C(x_1,x_2,x_3) \rightarrow \\ &\qquad (\exists y_1,y_2)\ P(y_1,y_2,y_2) \wedge P(y_1,x_2,x_3)\\ &C(x_1,x_2,x_3) \rightarrow (\exists y_1) P(y_1,x_3,x_2)\\ &C(x_1,x_2,x_2) \rightarrow Q(x_1)\}\end{aligned}$$

$$\begin{aligned}\Sigma'_{st} = \{&C(x_1,x_2,x_3) \rightarrow (\exists y_1)\ P(y_1,x_2,x_3)\\ &C(x_1,x_2,x_3) \rightarrow Q(x_1)\}\end{aligned}$$

$$\Sigma_t \ = \{P(x_1,x_2,x_3) \rightarrow x_2 = x_3\}$$

We have $\Sigma_{st} \cup \Sigma_t \equiv \Sigma'_{st} \cup \Sigma_t$. Moreover, both Σ_{st} and Σ'_{st} are in normal form w.r.t. the Rules 1 – 5 from Section 3. However, $\Sigma_{st} \not\equiv \Sigma'_{st}$ holds. □

In contrast, the equivalence of two sets of target egds is not influenced by the presence of s-t tgds, as the following lemma shows.

Lemma 9 *Suppose that $\Sigma = \Sigma_{st} \cup \Sigma_t$ and $\Upsilon = \Upsilon_{st} \cup \Upsilon_t$ are two logically equivalent sets of s-t tgds and target egds. Then, Σ_t and Υ_t are equivalent.*

Proof W.l.o.g. assume that there exists an $\varepsilon : \varphi(\mathbf{x}) \rightarrow \sigma(\mathbf{x}) \in \Upsilon_t$ s.t. $\Sigma_t \not\models \varepsilon$. That is, the set $L = At(\varphi(\mathbf{x}))^{\Sigma_t}$ of atoms of the antecedent of ε chased with Σ_t does not satisfy ε. However, it does satisfy Σ_t. Now, consider the pair of instances $\langle \emptyset, L\rangle$. Since $L \models \Sigma_t$, $\langle \emptyset, L\rangle \models \Sigma$ and $\langle \emptyset, L\rangle \not\models \Upsilon$, which is a contradiction. □

Recall that we are only considering *logical* equivalence of dependencies here. The study of weaker notions of equivalence [10] which only take attainable target instances into account (which is not the case for L in the above proof) has been initiated in [23].

In order to work with logical equivalence, we need a way to test logical implication of mappings. However, since we are now dealing with s-t tgds and egds, the declarative implication criterion from Lemma 1 no longer works. Instead, we take the chase-based procedure by Beeri and Vardi [4], applicable to any embedded dependencies that cannot cause an infinite chase (which is clearly the case when all tgds are s-t tgds).

Lemma 10 *[4] Let Σ be a set of acyclic tgds and egds and let δ be either a tgd or an egd. Let $\varphi(\mathbf{x})$ denote the antecedent of δ and let T denote the database obtained by chasing $At(\varphi(\mathbf{x}))$ with Σ. The variables in $\mathbf{x}$ are considered as labeled nulls. Then $\Sigma \models \delta$ iff $T \models \delta$ holds.*

Analogously to Rule 4 in Figure 1, we also need a rule for deleting redundant tgds in the presence of target egds. We shall refer to this rule as the Rule E1 in the rewrite rule system to be constructed in this section, which is specified in Figure 4. As in the tgd-only case, the primary goal of such a rewrite rule system is the definition of a unique normal form of the s-t tgds – but now taking also the target egds into account. The

```
Procedure PROPAGATE
Input: A set of s-t tgds and target egds $\Sigma = \Sigma_{st} \cup \Sigma_t$
Output: Sets of source egds $\Sigma_s$ and rewritten s-t tgds ${\Sigma_{st}}^*$
1. Set $\Sigma_s = {\Sigma_{st}}^* = \emptyset$;
2. for each s-t tgd $\tau\colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$ in $\Sigma_{st}$ do
  /* (a) non-frozen antecedent database */
     $I := At(\varphi(\mathbf{x}))$;
  /* (b) chase with $\Sigma = \Sigma_{st} \cup \Sigma_t$ */
     $\langle J_S, J_T \rangle := \langle I, \emptyset \rangle^{\Sigma}$;
  /* (c) transform s-t tgd $\tau$ into $\tau'$ */
     Let $J^* = core(J_T)$, whereby the terms occurring in $J_S$
                              are considered as constants.
     Let $\mathbf{y}$ be a tuple of all variables from $var(J_T) \setminus var(J_S)$;
     $\tau' := \left( \bigwedge_{A \in J_S} A \right) \rightarrow (\exists \mathbf{y}) \bigwedge_{B \in J^*} B$;
     ${\Sigma_{st}}^* := {\Sigma_{st}}^* \cup \{\tau'\}$;
  /* (d) generate source egds */
     Compute a substitution $\lambda$ s.t. $At(\varphi(\mathbf{x}\lambda)) = J_S$;
     for each pair of variables $x_j, x_k \in \mathbf{x}$ do
         if $x_j\lambda = x_k\lambda$ then
             $\Sigma_s := \Sigma_s \cup \{\varphi(\mathbf{x}) \rightarrow x_j = x_k\}$;
  end for;
```

Fig. 3 Procedure Propagate.

first step towards this goal is to incorporate the effects of egds into s-t tgds. As we have already pointed out in Section 1, this may require the introduction of source egds. Since we only consider source instances containing no variables (and not the recent semantics of [12]), there will be no source chase. The source egds are only meant to capture the failure conditions which cannot be detected otherwise after the rewriting of the s-t tgds.

In Figure 3, we present the procedure PROPAGATE, which incorporates, to some extent, the effect of the target egds into the s-t tgds and thereby possibly generates source egds. The idea of this procedure is that, for every s-t tgd τ, we identify all egds that will be applicable whenever τ is. Moreover, we want that all equalities enforced by these egds should already be enforced in the s-t tgd. Note that the chase in step 2.(b) is not the usual chase in data exchange. Here, in order to propagate backwards the effect of the target egds, in step 2.(b) we chase the database $I = At(\varphi(\mathbf{x}))$ with labeled nulls, which instantiate the variables from $\mathbf{x}$. We assume that this chase always succeeds: the only reason for failure on I could be constants occurring in s-t tgds of Σ. If this is the case, however, the chase is certain to fail on any source instance satisfying the antecedent of τ. Thus, such a τ can be simply replaced by a source egd of the form $\varphi(\mathbf{x}) \rightarrow \bot$ to rule out instances on which τ would fire.

Example 11 We now apply the PROPAGATE procedure to $\Sigma = \Sigma_{st} \cup \Sigma_t$ from Example 10. We start the loop in step 2 of the procedure with the first tgd of Σ_{st} τ: $C(x_1, x_2, x_3) \rightarrow (\exists y_1, y_2)\ P(y_1, y_2, y_2) \wedge P(y_1, x_2, x_3)$.

(a) $I := \{C(x_1, x_2, x_3)\}$. (We now consider every x_i as a labeled null).

(b) Chasing $\langle I, \emptyset \rangle$ with Σ_{st} yields $I' = \{C(x_1, x_2, x_3), P(y_1', y_2', y_2'), P(y_1', x_2, x_3), P(y_1'', x_3, x_2)\}$. The egd of Σ_t is then applied, resulting in $I'' = \{C(x_1, x_2, x_2), P(y_1', y_2', y_2'),\ P(y_1', x_2,\ x_2), P(y_1'', x_2, x_2)\}$. Note, that the egd application affected the "source" atom C. Now, the third tgd in Σ_{st} becomes applicable, producing the ultimate instance $\langle J_S, J_T \rangle = \langle I, \emptyset \rangle^{\Sigma} = \{C(x_1,\ x_2,\ x_2), P(y_1', y_2', y_2'),\ P(y_1', x_2, x_2), P(y_1'', x_2, x_2), Q(x_1)\}$.

(c) We got instances $J_S = \{C(x_1, x_2, x_2)\}$ and $J_T = \{P(y_1', y_2', y_2'),\ P(y_1', x_2, x_2), P(y_1'', x_2, x_2), Q(x_1)\}$. Core computation of J_T yields $J^* = \{P(y_1', x_2,\ x_2),\ Q(x_1)\}$. The s-t tgd τ is thus transformed into the following τ': $C(x_1, x_2, x_2) \rightarrow \exists y_1'\ P(y_1', x_2, x_2) \wedge Q(x_1)$.

(d) We compute the substitution $\lambda = \{x_3 \leftarrow x_2\}$, which maps the only atom $C(x_1, x_2, x_3)$ in $\varphi(\mathbf{x})$ onto instance $J_S = \{C(x_1,\ x_2, x_2)\}$. Hence, we get one source egd $C(x_1, x_2,\ x_3) \rightarrow x_2 = x_3$.

Finally, after the first iteration of the loop, we have ${\Sigma_{st}}^* = \{C(x_1, x_2, x_2) \rightarrow (\exists y_1')\, P(y_1', x_2, x_2) \wedge Q(x_1)\}$ and $\Sigma_s = \{C(x_1, x_2, x_3) \rightarrow x_2 = x_3\}$. The remaining two iterations do not change the tgds of Σ (and thus also introduce no further source egds). □

The PROPAGATE procedure never increases the size of the antecedents of s-t tgds. Hence, the cost of the join-operations when computing the canonical universal solution is not affected. On the other hand, the size of the conclusions is normally increased by this procedure. Note however that all atoms thus accumulated in the conclusion of some s-t tgd τ would be generated in a target instance anyway, whenever τ fires. We will ultimately discuss the deletion of redundant atoms from the conclusion of the s-t tgds (via a rule similar to Rule 5 from Figure 1). However, for the time being, it is convenient to have all these atoms present. This ensures that dependencies resulting from the PROPAGATE procedure possess the following essential properties.

Lemma 11 *Consider a set $\Sigma = \Sigma_{st} \cup \Sigma_t$ of s-t tgds Σ_{st} and target egds Σ_t. Moreover, let $(\Sigma_s, {\Sigma_{st}}^*)$ denote the result of* PROPAGATE(Σ_{st}, Σ_t)*. Then, the following conditions hold:*

(1) For every s-t tgd $\tau \in {\Sigma_{st}}^*$, let I_τ be a database obtained from the antecedent $\varphi(\mathbf{x})$ of τ by instantiating the variables of $\mathbf{x}$ with fresh distinct constants. Then, the chase of I_τ with $\Sigma_{st} \cup \Sigma_t$ is successful.

(2) For every source instance I: if $I \not\models \Sigma_s$ then the chase of I with $\Sigma_{st} \cup \Sigma_t$ fails.

Proof (1) After the successful completion of the chase in step 2.(b) of the PROPAGATE procedure, all necessary unifications in the antecedent relations have been performed. Hence, the instance $\langle At(\varphi(\mathbf{x})),\ At(\psi(\mathbf{x}, \mathbf{y})) \rangle$ for an s-t tgd $\varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$ in ${\Sigma_{st}}^*$ satisfies both Σ_{st} and Σ_t. Freezing the variables in $At(\varphi(\mathbf{x}))$ (i.e., taking them as constants) makes no difference.

(2) An inspection of steps 2.(b) and (d) of the Propagate procedure reveals that Σ_s enforces only those equalities which are implied by $\Sigma_{st} \cup \Sigma_t$. Therefore, a violation of Σ_s means that also $\Sigma_{st} \cup \Sigma_t$ is violated. □

Lemma 12 *The* Propagate *procedure is correct, i.e.: let $\Sigma = \Sigma_{st} \cup \Sigma_t$ and let $(\Sigma_s, {\Sigma_{st}}^*)$ result from a call of* Propagate(Σ_{st}, Σ_t). *Then $\Sigma \equiv \Sigma'$ for $\Sigma' = \Sigma_s \cup {\Sigma_{st}}^* \cup \Sigma_t$.*

Proof The Propagate procedure leaves the set Σ_t unchanged. Moreover, Lemma 11, part (2), implies $\Sigma \models \Sigma_s$. It thus remains to show $\Sigma \models \tau'$ for every $\tau' \in {\Sigma_{st}}^*$ and $\Sigma' \models \tau$ for every $\tau \in \Sigma_{st}$. These relationships are proved by inspecting the loop in Propagate (in particular, step 2.b) and checking that the implication criterion of [4] recalled in Lemma 10 is fulfilled.

$[\Sigma \models \tau']$ Let $\tau' \colon \varphi'(\mathbf{x}') \to (\exists \mathbf{y}')\psi'(\mathbf{x}', \mathbf{y}')$ be obtained by applying the loop of Propagate to some s-t tgd $\tau \in \Sigma_{st}$ with $\tau \colon \varphi(\mathbf{x}) \to (\exists \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$. The unifications applied to $\varphi(\mathbf{x})$ in order to get $\varphi'(\mathbf{x}')$ are precisely the ones enforced by the chase of $At(\varphi(\mathbf{x}))$ with Σ in step 2.(b) of Propagate. Therefore, chasing $At(\varphi'(\mathbf{x}'))$ with Σ yields the same result as the chase of $I = At(\varphi(\mathbf{x}))$ with Σ, namely I^{Σ}. Hence, the conjunction of the atoms in the set J_T in step 2.(b) is satisfied by I^{Σ}. Now consider the conclusion $\psi'(\mathbf{x}', \mathbf{y}')$ of τ', which is obtained via core computation from J_T. $\psi'(\mathbf{x}', \mathbf{y}')$ is the conjunction of a subset of J_T, which is clearly also satisfied by I^{Σ}.

$[\Sigma' \models \tau]$ Let $\tau \colon \varphi(\mathbf{x}) \to (\exists \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$ be an s-t tgd in Σ_{st} and let $\tau' \colon \varphi'(\mathbf{x}') \to (\exists \mathbf{y}')\psi'(\mathbf{x}', \mathbf{y}')$ denote the result of applying the loop of Propagate to τ. Consider the (non-frozen) antecedent database $At(\varphi(\mathbf{x}))$ of τ. Chasing I with Σ' comes down to enforcing Σ_s (which transforms $At(\varphi(\mathbf{x}))$ into $At(\varphi'(\mathbf{x}'))$) followed by chasing $At(\varphi'(\mathbf{x}'))$ with τ'. The result of this chase is $I^* = At(\varphi'(\mathbf{x}')) \cup At(\psi'(\mathbf{x}', \mathbf{y}'))$. Note that $At(\psi'(\mathbf{x}', \mathbf{y}'))$ is the core of the chase of $At(\varphi(\mathbf{x}))$ with Σ. Hence, I^* must satisfy τ, from which the claim follows, by Lemma 10. □

The following property is easy to see and will be helpful subsequently.

Lemma 13 *Let $\Sigma = \Sigma_{st} \cup \Sigma_t$, and let $(\Sigma_s, {\Sigma_{st}}^*)$ denote the result of* Propagate(Σ_{st}, Σ_t). *Moreover, let $\tau \in {\Sigma_{st}}^*$ with $\tau \colon \varphi(\mathbf{x}) \to \exists(\mathbf{y})\ \psi(\mathbf{x}, \mathbf{y})$, and let I be a source instance with $I \subseteq At(\varphi(\mathbf{x}))$, such that elements of $\mathbf{x}$ are instantiated with distinct fresh constants in I.*

Then the chase of I both with $\Sigma = \Sigma_{st} \cup \Sigma_t$ and with $\Sigma^* = \Sigma_s \cup {\Sigma_{st}}^* \cup \Sigma_t$ succeeds. Moreover, $core(I^{\Sigma}) = core(I^{\Sigma^*})$.

Proof By condition (1) of Lemma 11, the chase with Σ and with Σ^* succeeds on the frozen antecedent database I_τ of τ. Hence, the chase must also succeed for any subset of I_τ. The equality $core(J^{\Sigma}) = core(J^{\Sigma^*})$ immediately follows from the correctness of the Propagate procedure, see Lemma 12. □

Splitting in the presence of egds. The following example illustrates that the splitting rule (i.e., Rule 3 in Figure 1) does not suffice to detect the possibility of splitting a "bigger" tgd into smaller ones in the presence of target egds:

Example 12 Consider the following mapping $\Sigma = \{\tau, \epsilon\}$

$\tau\colon\ S(x, z_1) \wedge S(x, z_2) \to R(z_1, y) \wedge Q(z_2, y)$
$\epsilon\colon\ R(x_1, y_1) \wedge Q(x_2, y_2) \to y_1 = y_2$

It is easy to check that Σ is equivalent to the mapping $\Sigma' = \{\tau_1, \tau_2, \epsilon\}$ with the same target egd and two s-t tgds, each containing only a subset of the antecedent and conclusion atoms of τ:

$\tau_1\colon\ S(x, z_1) \to R(z_1, y)$
$\tau_2\colon\ S(x, z_2) \to Q(z_2, y)$

The Rule 3 from Section 3 does not allow such a splitting, however. □

In some sense, the splitting in the above example still had significant similarities with splitting in the absence of egds, namely: the basic idea of distributing the conclusion atoms over several dependencies is still present when target egds have to be taken into account. However, we have to deal with a significant extension here: Without egds it would never be possible to split the connected component (w.r.t. the existential variables) of the conclusion of a tgd. As we have seen in the above example, egds may allow us to tear a connected component apart. Moreover, splitting in the presence of egds is not merely distributing atoms of the conclusion of some dependency over several ones. The following example illustrates that we may also have to copy atoms in order to further split the conclusion of a tgd.

Example 13 Consider the following mapping Σ

$\tau : S(x_1, x_2) \wedge S(x_1, x_3) \to$
$\quad R(x_2, y) \wedge P(y, x_2) \wedge Q(y, x_3)$
$\epsilon : R(x, y_1) \wedge R(x, y_2) \to y_1 = y_2$

The s-t tgd τ can be rewritten in the following way:
$\tau_1 : S(x_1, x_2) \wedge S(x_1, x_3) \to R(x_2, y) \wedge Q(y, x_3)$
$\tau_2 : S(x_1, x_2) \to R(x_2, y) \wedge P(y, x_2)$

Both τ_1 and τ_2 must contain an R-atom. □

We observe that the total number of atoms in all conclusions in the resulting mapping in Example 13 has increased. But compared with the original mapping τ, each conclusion is strictly smaller than the original one. (i.e., is obtained by deletion of at least one atom and possibly the renaming of some variable occurrences). We thus generalize the notion of *split-reduced* mappings from the s-t tgd-only case:

Definition 10 Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ be a mapping. The tgd $\tau\colon \varphi(\bar{x}, \bar{z}) \to \psi(\bar{x}, \bar{y}) \in \Sigma_{st}$ is *egd-split-reduced*, if it is not possible to replace it by a set of new s-t tgds τ_i with antecedent φ_i and conclusion ψ_i, s.t. $At(\varphi_i) \subseteq At(\varphi)$ and $|At(\psi_i)| < |At(\psi)|$. Σ_{st} is said to be egd-split-reduced iff each dependency in it is.

The above notion of egd-split-reduced mappings generalizes the notion of split-reduced mappings from Definition 3 to mappings with target egds. The connection between the two notions of splitting is formalized by the following lemma:

Lemma 14 *Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ with $\Sigma_t = \emptyset$ and suppose that Σ_{st} cannot be simplified by any of the Rules 1,4, and 5 from Figure 1 (i.e., the rules which would reduce $ConSize(\Sigma_{st})$ are not applicable). Then the following equivalences hold: Σ_{st} is egd-split-reduced iff Σ_{st} is split-reduced iff Rule 3 (i.e., splitting) cannot be applied.*

Proof The second equivalence was already shown in Lemma 8. Below we show that Σ_{st} is egd-split-reduced iff Rule 3 (i.e., splitting) cannot be applied.

First suppose that Rule 3 (i.e., splitting) actually can be applied to Σ_{st}. Then some $\tau \in \Sigma_{st}$ can be replaced by tgds $\tau_1, \ldots, \tau_n$, s.t. the antecedent of each τ_i coincides with the antecedent of τ and the conclusion of each τ_i is a proper subset of the conclusion of τ. Hence, Σ_{st} is not egd-split-reduced.

Now suppose that Σ_{st} is not egd-split-reduced. We have to show that then Rule 3 can be applied. Suppose to the contrary that Rule 3 cannot be applied. We derive a contradiction by showing that then one of the Rules 1, 4, 5 is applicable to Σ: Since Σ_{st} is not egd-split-reduced, there exists a $\tau \in \Sigma_{st}$ with antecedent φ which can be replaced by a set of new tgds $\{\tau_1, \ldots, \tau_n\}$, s.t. for every i, $At(\varphi_i) \subseteq At(\varphi)$ and $|At(\psi_i)| < |At(\psi)|$ hold, where φ_i and ψ_i respectively denote the antecedent and conclusion of τ. Moreover, $\Sigma \equiv \Sigma'$ holds with $\Sigma' = (\Sigma \setminus \{\tau\}) \cup \{\tau_1, \ldots, \tau_n\}$. In particular, $\Sigma' \models \tau$. By Lemma 4, then either (1) $\Sigma' \models \tau'$ holds for some proper instance τ' of τ (see Definition 7) or (2) τ is already implied by a single tgd $\sigma \in \Sigma'$.

In case (1), we clearly also have $\Sigma \models \tau'$ for the proper instance τ' of τ. But then, by Lemma 5, Rule 5 is applicable to τ, which is a contradiction. It remains to consider case (2). Clearly, $\sigma \notin \Sigma \setminus \{\tau\}$ since otherwise τ could be deleted from Σ via Rule 4. So let $\sigma = \tau_j$ for some j. We thus have $\bar{\Sigma} \models \tau$ with $\bar{\Sigma} = (\Sigma \setminus \{\tau\}) \cup \tau_j$ and, therefore, also $\bar{\Sigma} \equiv \Sigma$. Moreover, $|At(\psi_j)| < |At(\psi)|$ holds, which implies $ConSize(\bar{\Sigma}) < ConSize(\Sigma)$. Now suppose that we transform both Σ and $\bar{\Sigma}$ into the unique (up to isomorphism) normal form Σ^* according to Definition 8. By assumption, none of the Rules 1, 3, 4, and 5 is applicable to Σ. Hence, by the same considerations as in the proof of Lemma 8, Σ^* is obtained by successive applications of Rule 2, which leaves the conclusions of the tgds unchanged. Hence, we have $ConSize(\Sigma^*) = ConSize(\Sigma)$. On the other hand, if we transform $\bar{\Sigma}$ into the normal form Σ^*, then we never increase the conclusion size. Hence, from the inequality $ConSize(\bar{\Sigma}) < ConSize(\Sigma)$ we may infer $ConSize(\Sigma^*) < ConSize(\Sigma)$, which contradicts the equality that we have just derived. □

In Figure 4 we present the Rule ES, whose exhaustive application obviously transforms any mapping into an egd-split-reduced one. Alas, the following example shows that we cannot hope to get a unique egd-split-reduced mapping.

Example 14 Consider the schema mapping Σ consisting of a single s-t tgd and a number of target egds:

$$S(1,x) \wedge S(1,2) \wedge S(y,2) \to T(x,y,z) \wedge P(x,z) \wedge R(y,z) \wedge Q(z,v,w)$$

$$T(x,y,z) \wedge P(x,z) \wedge Q(z,v,w) \to v = w$$
$$T(x,y,z) \wedge R(y,z) \wedge Q(z,v,w) \to v = w$$

$$T(x,y,z_1) \wedge P(x,z_2) \wedge Q(w,v,v) \to z_1 = z_2$$
$$T(x,y,z_1) \wedge R(y,z_2) \wedge Q(w,v,v) \to z_1 = z_2$$

This mapping is not in the egd-split-reduced form, since the antecedent of the s-t tgd can be shrunk by extracting either the P or the Q atom from the conclusion. Σ'_{st} and Σ''_{st} are two possible transformations of Σ into egd-split-reduced form via the Rule ES.

$$\Sigma'_{st} = \{\ S(1,x) \wedge S(1,2) \wedge S(y,2) \to T(x,y,z) \wedge R(y,z) \wedge Q(z,v,w),\ S(1,x) \wedge S(1,2) \to P(x,z)\} \text{ and}$$

$$\Sigma''_{st} = \{\ S(1,x) \wedge S(1,2) \wedge S(y,2) \to T(x,y,z) \wedge P(x,z) \wedge Q(z,v,w),\ S(1,2) \wedge S(y,2) \to R(y,z)\}$$ □

Clearly, the problem in Example 14 is not just due to the definition of the Rule ES. Instead, it is an intrinsic problem of the notion of egd-split-reduced mappings. Apparently this extent of splitting is too strong. We shall therefore relax the notion of egd-split-reduced. This will be the topic of the next paragraph.

Antecedent-split-reduced mappings. In Example 14 we observed that certain antecedent atoms may be freely distributed between several tgds, if the idea of splitting from Section 3 is directly adopted in the setting with target constraints. Therefore, in order to arrive at an intuitive definition of a unique normal form, we shift our focus to the antecedents:

Definition 11 Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ be a mapping. The s-t tgd $\tau\colon \varphi(\bar{x}, \bar{z}) \to \psi(\bar{x}, \bar{y}) \in \Sigma_{st}$ is *antecedent-split-reduced*, if it is not possible to replace it with a set of

Rewrite Rules in the Presence of Egds

Rule E1 (General implication).
$\Sigma \Longrightarrow \Sigma \setminus \{\tau\}$
if $\Sigma \setminus \{\tau\} \models \tau$.

Rule E2 (Restriction of an antecedent to endomorphic images).
$\Sigma \Longrightarrow (\Sigma \setminus \{\tau\}) \cup \{\tau_1, \ldots, \tau_n\}$
if $\tau \colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$
and $(\Sigma \setminus \{\tau\}) \cup \{\tau_1, \ldots, \tau_n\} \models \tau$
and for each $i \in \{1, \ldots, n\}$
$\tau_i \colon \varphi_i(\mathbf{x}_i) \rightarrow (\exists \mathbf{y}_i)\psi_i(\mathbf{x}_i, \mathbf{y}_i)$,
s.t. $\exists \lambda$, $At(\varphi(\mathbf{x}\lambda)) = At(\varphi_i(\mathbf{x}_i)) \subset At(\varphi(\mathbf{x}))$
and $\psi_i(\mathbf{x}_i, \mathbf{y}_i) = \mathit{core}(At(\varphi_i(\mathbf{x}_i))^{\Sigma})$.

Rule E3 (Implication of atoms in the conclusion).
$\Sigma \Longrightarrow (\Sigma \setminus \{\tau\}) \cup \{\tau'\}$
if $\tau \colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$
and $\tau' \colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y}')\psi'(\mathbf{x}, \mathbf{y}')$,
s.t. $At(\psi'(\mathbf{x}, \mathbf{y}')) \subset At(\psi(\mathbf{x}, \mathbf{y}))$
and $(\Sigma \setminus \{\tau\}) \cup \{\tau'\} \models \tau$.

Rule ES (generalized splitting in the presence of egds).
$\Sigma \Longrightarrow (\Sigma \setminus \{\tau\}) \cup \{\tau_1, \ldots, \tau_n\}$
if $\tau \colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$
and $(\Sigma \setminus \{\tau\}) \cup \{\tau_1, \ldots, \tau_n\} \models \tau$
and for each $i \in \{1, \ldots, n\}$
$\tau_i \colon \varphi_i(\mathbf{x}_i) \rightarrow (\exists \mathbf{y}_i)\psi_i(\mathbf{x}_i, \mathbf{y}_i)$,
s.t. $\emptyset \subset At(\varphi_i(\mathbf{x}_i)) \subseteq At(\varphi(\mathbf{x}))$
and $At(\psi_i(\mathbf{x}_i, \mathbf{y}_i\mu_i)) \subset At(\psi(\mathbf{x}, \mathbf{y}))$ for a substitution μ_i.

Rule E2-eager (Restriction of an antecedent to subsets).
$\Sigma \Longrightarrow (\Sigma \setminus \{\tau\}) \cup \{\tau_1, \ldots, \tau_n\}$
if $\tau \colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x}, \mathbf{y})$
and $(\Sigma \setminus \{\tau\}) \cup \{\tau_1, \ldots, \tau_n\} \models \tau$
and for each $i \in \{1, \ldots, n\}$
$\tau_i \colon \varphi_i(\mathbf{x}_i,) \rightarrow (\exists \mathbf{y}_i)\psi_i(\mathbf{x}_i, \mathbf{y}_i)$,
s.t. $\emptyset \subset At(\varphi_i(\mathbf{x}_i)) \subset At(\varphi(\mathbf{x}))$
and $\psi_i(\mathbf{x}_i, \mathbf{y}_i) = \mathit{core}(At(\varphi_i(\mathbf{x}_i))^{\Sigma})$.

Fig. 4 Rewrite rules in the presence of egds.

new s-t tgds τ_i each having strictly smaller antecedent, i.e., for the antecedents φ_i, we get $|At(\varphi_i)| < |At(\varphi)|$. Σ_{st} is said to be antecedent-split-reduced iff each dependency in it is.

In order to transform a mapping into an antecedent-split-reduced one, we define the rule E2-eager in Figure 4. It can be shown that any normal form under a rule rewrite system containing Rule E2-eager is antecedent-split-reduced and vice versa. In this rule, we have to inspect all subsets of the antecedent database of each tgd. Actually, we will show that it suffices to check all subsets φ_i of an antecedent $\varphi(\mathbf{x})$, such that φ_i is a proper endomorphic image of $\varphi(\mathbf{x})$. This is what the Rule E2 in Figure 4 does. Clearly, the number of endomorphic images is, in general, far smaller than the number of all subsets. In particular, we never have to check antecedents smaller than the core of the already present antecedents (here we mean the core of the conjunctive query – without distinguishing two groups of variables as we did in the definition of the Rules 1 and 2). The following Theorem shows that both, the rule E2-eager and the rule E2 exactly capture the notion of antecedent-split-reduced mappings.

Theorem 5 *Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ be a schema mapping in which no tgd can be deleted via the Rule E1. Then the following properties are equivalent:*

1. *Σ is antecedent-split-reduced.*
2. *Σ is reduced w.r.t. Rule E2-eager.*
3. *Σ is reduced w.r.t. Rule E2.*

Proof (1) ⇔ (2) follows directly from the definition of antecedent-split-reduced form.

(2) ⇒ (3) is trivial, as every proper endomorphic image of a set of atoms $At(\varphi)$ is a proper subset of $At(\varphi)$.

(3) ⇒ (2). Consider an application of the rule E2-eager, in which it substitutes some s-t tgd τ in Σ with a set of s-t tgds T, s.t. $\Sigma \equiv \Sigma \setminus \{\tau\} \cup T$. Let now φ be the antecedent of τ, and φ_i be the antecedent of some tgd $\tau_i \in T$. By definition of E2-eager, $At(\varphi_i) \subset At(\varphi)$. We have to show that $At(\varphi_i)$ is an endomorphic image of $At(\varphi)$. Suppose to the contrary that it is not. We show that then τ_i is "superfluous" in T: Namely, the property $(\Sigma \setminus \{\tau\}) \cup T \setminus \{\tau_i\} \equiv (\Sigma \setminus \{\tau\}) \cup T$ holds. Of course, if T only contains such tgds which are superfluous, then the tgd τ itself can be deleted by the E1 rule, which is a contradiction to the assumption of this theorem. On the other hand, if T is non-empty and contains only tgds whose antecedent is an endomorphic image of φ then also the E2 Rule is applicable.

To show that τ_i is superfluous in T, consider the following two cases:

(a) There exists no homomorphism $\varphi \rightarrow \varphi_i$. Then, τ_i is superfluous in T, in a sense that the property $(\Sigma \setminus \{\tau\}) \cup T \setminus \{\tau_i\} \models \tau$ holds. Indeed, according to E2-eager, the conclusion of τ_i was created by chasing $At(\varphi_i)$ with Σ. Since $\varphi \not\rightarrow \varphi_i$, τ played no role in that chase, and thus $\Sigma \setminus \{\tau\} \models \tau_i$ holds, and hence τ_i is indeed superfluous.

(b) There exists a homomorphism $\varphi \rightarrow \varphi_i$. Let Λ denote the set of all homomorphisms $\varphi \rightarrow \varphi_i$. Then we define $T_\varphi \subset T$, s.t. $T_\varphi = \{\tau_j \in T \mid At(\varphi_j) = \varphi\lambda$ for some $\lambda \in \Lambda\}$, i.e., the antecedents of the tgds in T_φ are subsets of φ_i and, at the same time, endomorphic images of φ. Similarly to the previous case, one can show that $(\Sigma \setminus \{\tau\}) \cup T_\varphi \models \tau_i$ holds. Thus τ_i is superfluous.

By construction of τ_i, we have $\Sigma \models \tau_i$. Indeed, consider the implication test of Lemma 10, in which the database $At(\varphi_i)$, obtained from the antecedent of τ_i, is chased by Σ. The effect of τ in this chase is exactly the generation of conclusion atoms by instantiating the existentially quantified variables $\mathbf{y}$ in $\psi(\mathbf{x}\lambda, \mathbf{y})$, where $\psi(\mathbf{x}, \mathbf{y})$ is the conclusion of τ and λ is an endomorphism sending the antecedent $\varphi(\mathbf{x})$ of τ on its part, which is the antecedent of τ_i. But then, by definition of E2, there exists a substitution μ such that $\psi(\mathbf{x}\lambda_i, \mathbf{y}\mu)$ is a subformula of the conclusion of some $\tau_\varphi \in T_\varphi$: Indeed, E2 takes exactly all endomorphic images of φ and performs the chase with Σ to derive the conclusion of an s-t tgd in T_φ. Note that the substitution μ captures the effect of egds possibly fired by the chase which derives the conclusion of τ_φ from its antecedent $\varphi\lambda$; these egds will surely be also fired in the chase of $At(\varphi_i)$.

Let the chase of $At(\varphi_i)$ with τ yield the target instance J, and the chase of $At(\varphi_i)$ with T_φ yield J'. Then, it is easy to see that a homomorphism $J \to J'$ must exist. But then, $At(\varphi_i)^\Sigma \models \tau_i$ only if $At(\varphi_i)^{\Sigma'} \models \tau_i$ with $\Sigma' = (\Sigma \setminus \{\tau\}) \cup T_\varphi$ Hence, it must be the case that $(\Sigma \setminus \{\tau\}) \cup T_\varphi$ holds and, therefore, τ_i is superfluous. □

There is a close connection between antecedent-split-reduced mappings and the split-reduced form from Definition 3:

Lemma 15 *Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ with $\Sigma_t = \emptyset$ and suppose that Σ_{st} cannot be simplified by any of the Rules 1,4, and 5 from Figure 1 (i.e., the rules which would reduce ConSize(Σ_{st}) are not applicable). Then the following equivalence holds: Σ_{st} is antecedent-split-reduced iff Rule 3 (i.e., splitting) cannot be applied in such a way that the antecedents of all resulting dependencies can be further simplified (by Rule 2).*

Proof First suppose that Rule 3 (i.e., splitting) followed by a simplification of the antecedent of each new tgd is applicable. Then, in the first place, some $\tau \in \Sigma_{st}$ can be replaced by tgds $\tau_1, \dots, \tau_n$, s.t. the antecedent of each τ_i coincides with the antecedent of τ and the conclusion of each τ_i is a proper subset of the conclusion of τ. Moreover, each τ_i can then be transformed via Rule 2 into τ_i', s.t. the antecedent of τ_i' is a proper subset of the antecedent of τ and, therefore, also of τ. Hence, Σ_{st} is not antecedent-split-reduced.

For the opposite direction, suppose that Σ_{st} is not antecedent-split-reduced. We have to show that then Rule 3 can be applied followed by applying Rule 2 to each of the new tgds. Since Σ_{st} is not antecedent-split-reduced, there exists a $\tau \in \Sigma_{st}$ with antecedent φ which can be replaced by a set of new tgds $\{\tau_1, \dots, \tau_n\}$, s.t. for every i, $At(\varphi_i) \subset At(\varphi)$ and $|At(\psi_i)| < |At(\psi)|$ hold, where φ_i and ψ_i respectively denote the antecedent and conclusion of τ. Moreover, $\Sigma \equiv \Sigma'$ holds with $\Sigma' = (\Sigma \setminus \{\tau\}) \cup \{\tau_1, \dots, \tau_n\}$.

Analogously to the proof of Theorem 5, we may assume w.l.o.g., that each of the new antecedents φ_i is an endomorphic image of φ. Moreover, we may assume w.l.o.g., that each ψ_i contains only one connected component since otherwise we simply split τ_i further via Rule 3. We claim that for every connected component of ψ, there is one i, s.t. this connected component corresponds to ψ_i. Suppose to the contrary that there is a connected component χ of ψ which does not have a corresponding ψ_i. Then we derive a contradiction as follows. The tgd τ' obtained from τ by reducing the conclusion ψ to χ is clearly implied by Σ'. Hence, by Lemma 4, either (1) $\Sigma' \models \tau''$ holds for some proper instance τ'' of τ' (see Definition 7) or (2) τ' is already implied by a single tgd $\sigma \in \Sigma'$. In case (1), we thus have $\Sigma \models \tau''$ for the proper instance τ'' of τ'. But then, also $\Sigma \models \tau^*$, where τ^* denotes the tgd obtained from τ by replacing the connected component χ by the conclusion of τ'' (i.e., a proper instance of χ) and leaving all other connected components unchanged. By Lemma 5, Rule 5 is applicable to $\tau \in \Sigma$, which is a contradiction. Now consider case (2), i.e., τ' is implied by a single tgd $\sigma \in \Sigma'$. Clearly, σ cannot be contained in $\Sigma \setminus \{\tau\}$ since this would mean that the connected component χ of the conclusion of τ could be deleted from Σ via Rule 5. So suppose that $\sigma = \tau_j$ for some j, i.e., we have $\tau_j \models \tau'$. By Lemma 10, this means that the conclusion χ of τ' can be obtained by chasing the antecedent φ of τ' with χ. Note however that χ is a single connected component. Hence, all of χ is obtained in a single chase step, since otherwise we conclude that also a proper instance of τ' is implied by τ_j and we proceed as in case (1). Since χ is obtained in a single chase step, the conclusion of τ_j indeed comprises all of χ.

To conclude the proof, recall the above observation that each of the antecedents φ_i is an endomorphic image of φ. But then we can indeed apply the Rule 2 in the reverse direction to extend each φ_i to φ. Let the resulting tgds be called $\{\bar{\tau}_1, \dots, \bar{\tau}_n\}$. Then we indeed have that $\tau \in \Sigma$ may be replaced by $\{\bar{\tau}_1, \dots, \bar{\tau}_n\}$ via Rule 3 and each $\bar{\tau}_i$ may be further simplified via Rule 2 to τ_i with strictly smaller antecedent. □

Most importantly, the notion of antecedent-split-reduced mappings allows us to define a unique (up to isomorphism) normal form of the set of s-t tgds. To this end, we consider the transformation of an arbitrary mapping consisting of s-t tgds and target egds by the Propagate procedure from Figure 3 followed by exhaustive application of the rules E1 and E2 from Figure 4. Below we show that the resulting normal form is indeed unique up to isomorphism:

Lemma 16 *The rewrite rules E1 and E2 in Figure 4 are correct, i.e.: Let Σ be a set of dependencies and let Σ' be the result of applying one of the rules E1 or E2 to Σ. Then $\Sigma \equiv \Sigma'$.*

Proof The correctness follows directly from the fact that a logical implication test is built into the rules E1 and E2. □

Theorem 6 *Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ and $\Upsilon = \Upsilon_{st} \cup \Upsilon_t$ be two logically equivalent equivalent sets consisting of s-t tgds and target egds and let $\langle \Sigma_s, {\Sigma_{st}}^*, \Sigma_t \rangle$ and $\langle \Upsilon_s, \Upsilon_{st}^*, \Upsilon_t \rangle$ be obtained from Σ respectively Υ by first applying the Propagate procedure and then exhaustively applying the rules E1 and E2 to these mappings. Then $\Sigma_t \equiv \Upsilon_t$ holds and ${\Sigma_{st}}^*$ and Υ_{st}^* are isomorphic.*

Proof The equivalence $\Sigma_t \equiv \Upsilon_t$ was shown in Lemma 9. It remains to show that ${\Sigma_{st}}^*$ and Υ^*_{st} are isomorphic.

Let $\sigma \in {\Sigma_{st}}^*$ be an arbitrary s-t tgd in ${\Sigma_{st}}^*$. We have to show that it has an isomorphic analogue in Υ^*_{st} (and vice versa). Let $\bar{\Sigma}_\sigma$ denote the set of s-t tgds whose antecedents are the proper subsets of σ and whose conclusions are obtained by chasing the corresponding antecedent database with ${\Sigma_{st}}^* \cup \Sigma_t$ (i.e., we get s-t tgds analogous to the τ_i's in Rule E2). By Lemma 13, this is the same as chasing these particular source instances with Σ.

By $\Upsilon \models \sigma$, there exists a subset $T \subseteq \Upsilon_s \cup \Upsilon^*_{st} \cup \Upsilon_t$, s.t. $T \cup \Sigma_s \cup \bar{\Sigma}_\sigma \cup \Sigma_t \cup {\Sigma_{st}}^* \setminus \{\sigma\} \models \sigma$. We claim that there even exists a set $T_\sigma \subseteq \Upsilon^*_{st}$ with $T_\sigma \cup \Sigma_s \cup \bar{\Sigma}_\sigma \cup \Sigma_t \cup {\Sigma_{st}}^* \setminus \{\sigma\} \models \sigma$, s.t. every $\tau \in T_\sigma$ fulfills the following properties:

1. The antecedents $\varphi_\tau(\mathbf{x}_\tau)$ of τ and $\varphi_\sigma(\mathbf{x}_\sigma)$ of σ are homomorphically equivalent;
2. there exists a substitution λ, such that $\varphi_\tau(\mathbf{x}_\tau\lambda) = \varphi_\sigma(\mathbf{x}_\sigma)$. That is, the antecedent of τ can be mapped onto the entire antecedent of σ;
3. τ is not equivalent to any dependency in $\Sigma_{st} \setminus \{\sigma\}$

In order to prove this claim, we start with a set $T \subseteq \Upsilon_s \cup \Upsilon^*_{st} \cup \Upsilon_t$, s.t. $T \cup \Sigma_s \cup \bar{\Sigma}_\sigma \cup \Sigma_t \cup {\Sigma_{st}}^* \setminus \{\sigma\} \models \sigma$ and remove all parts from T until a subset $T_\sigma \subseteq T$ with the desired properties is obtained. It is convenient to write Σ^* as a short-hand for $\Sigma_s \cup \bar{\Sigma}_\sigma \cup \Sigma_t \cup {\Sigma_{st}}^* \setminus \{\sigma\}$.

(a) *Eliminate Υ_s from T.* This is justified by the fact that $\Upsilon^*_{st} \cup \Upsilon_t \models \sigma$ holds. Suppose to the contrary that this fact does not hold: that is, let I be an instance over the schema $\mathbf{S} \cup \mathbf{T}$, in which the only non-empty relations are those of the antecedent database $At(\varphi_\sigma(\mathbf{x}_\sigma))$ of σ. Then, chasing I with $\Upsilon_{st} \cup \Upsilon_t$ leads to an instance $I^{\Upsilon_{st} \cup \Upsilon_t} \not\models \sigma$, whereas $I^{\Upsilon_s \cup \Upsilon_{st} \cup \Upsilon_t} \models \sigma$. Since source dependencies in Υ_s are only applicable to relations of the source schema, it must hold that Υ_s modifies $At(\varphi_\sigma(\mathbf{x}_\sigma))$; otherwise there would be no difference between the two chase results. That is, $At(\varphi_\sigma(\mathbf{x}_\sigma)) \not\models \Upsilon_s$. By Lemma 11, part (2), this means that the chase of $At(\varphi_\sigma(\mathbf{x}_\sigma))$ with Υ fails. Thus, also the chase with Σ fails, which contradicts Lemma 11, part (1).

(b) *Eliminate Υ_t from T.* The correctness of this step follows immediately from the equivalence $\Upsilon_t \equiv \Sigma_t$ that we showed in Lemma 9.

(c) *Eliminate every tgd τ from T which is equivalent to some $\sigma' \in {\Sigma_{st}}^* \setminus \{\sigma\}$.* Clearly, after such a reduction, we still have $T \cup \Sigma^* \models \sigma$ with $\Sigma^* = \Sigma_s \cup \bar{\Sigma}_\sigma \cup \Sigma_t \cup {\Sigma_{st}}^* \setminus \{\sigma\}$.

(d) *Eliminate from T all dependencies with the antecedent $\varphi_i(\mathbf{x}_i)$ which is not homomorphically equivalent to the antecedent $\varphi_\sigma(\mathbf{x}_\sigma)$ of σ.* Indeed, for every s-t tgd $\tau_i \in \Upsilon^*_{st}$ with the antecedent "more specific" than $\varphi_\sigma(\mathbf{x}_\sigma)$, we may conclude that for arbitrary Σ', such that $\Sigma' \models \sigma$, it holds that $\Sigma' \setminus \{\tau_i\} \models \sigma$. For every τ_j with the antecedent "more general" than $\varphi_\sigma(\mathbf{x}_\sigma)$, we have that $\Sigma^* \setminus \{\sigma\} \models \tau_j$, and therefore, τ_j is redundant in $T \cup \Sigma^* \setminus \{\sigma\}$.

(e) *Eliminate from T all s-t tgds with the antecedents $\varphi_k(\mathbf{x}_k)$ such that there exists no variable substitution λ: $\varphi_k(\mathbf{x}_k\lambda) = \varphi_\sigma(\mathbf{x}_\sigma)$*, where $\varphi_\sigma(\mathbf{x}_\sigma)$ again denotes the antecedent of σ. First observe that there are no dependencies in T whose antecedents under any variable substitution are supersets of $\varphi_\sigma(\mathbf{x}_\sigma)$, since they are "more specific" than $\varphi_\sigma(\mathbf{x}_\sigma)$ and have therefore been removed in the previous step.

Now consider the substitutions λ_{ki}: $\varphi_k(\mathbf{x}_k\lambda_{ki}) \subset \varphi_\sigma(\mathbf{x}_\sigma)$ and the corresponding s-t tgds $\tau_k \in T$. We claim that the following property holds:

For any set of dependencies K such that $\tau_k \in K$, $K \models \sigma$ iff $(K \setminus \{\tau_k\}) \cup K_{\tau_k} \models \sigma$, where K_{τ_k} is the set of all instantiations of τ_k with λ_{ki}: $\tau_{ki} = \varphi_k(\mathbf{x}_k\lambda_{ki}) \rightarrow \exists \mathbf{y}_k\ \psi(\mathbf{x}_k\lambda_{ki}, \mathbf{y}_k)$.

The claim follows from the consideration of the implication test by Beeri and Vardi [4]: to chase the antecedent database $At(\varphi_\sigma(\mathbf{x}_\sigma))$ of σ, τ_k is instantiated by every λ_{ki} and thus has the same effect in the chase as K_{τ_k}. Hence, every τ_k in T whose antecedent cannot be projected onto the entire $\varphi_\sigma(\mathbf{x}_\sigma)$ may be replaced by the respective instantiations K_{τ_k}.

We now recall that the antecedents of the s-t tgds $\rho_l \in \bar{\Sigma}_\sigma \subset \Sigma^*$ range over all possible subsets of $\varphi_\sigma(\mathbf{x}_\sigma)$. That is, for each τ_{ki} with the antecedent $\varphi_k(\mathbf{x}_k\lambda_{ki})$ there exists ρ_{ki} with the identical antecedent and with the conclusion obtained by chasing $\varphi_k(\mathbf{x}_k\lambda_{ki})$ with Σ. Since Σ and Υ are equivalent, we conclude that $\rho_{ki} \models \tau_{ki}$, and thus $\Sigma^* \models K_{\tau_k}$ for every τ_k. Hence, it is indeed allowed to eliminate from T all s-t tgds with the antecedents $\varphi_k(\mathbf{x}_k)$ such that there exists no variable substitution λ: $\varphi_k(\mathbf{x}_k\lambda) = \varphi_\sigma(\mathbf{x}_\sigma)$.

After the above five elimination steps, T is indeed reduced to a set T_σ of the desired form. Note that T_σ is non-empty. This can be seen as follows: The s-t tgd σ is reduced w.r.t. rules E1 and E2. Hence, $\Sigma_s \cup \bar{\Sigma}_\sigma \cup \Sigma_t \cup {\Sigma_{st}}^* \setminus \{\sigma\} \not\models \sigma$ and, therefore, T_σ must be non-empty.

By obvious symmetry reasons, the same holds for any s-t tgd $\tau \in \Upsilon^*_{st}$ as well: each τ must also have such a corresponding non-empty set $S_\tau \subseteq {\Sigma_{st}}^*$, with elements satisfying the conditions 1–3.

We now construct a directed bipartite graph $G = (V_1, V_2, E)$ as follows: We associate the s-t tgds in ${\Sigma_{st}}^*$ and Υ^*_{st} with the vertices, s.t. $V_1 = {\Sigma_{st}}^*$ and $V_2 = \Upsilon^*_{st}$. Moreover, whenever $\tau \in T_\sigma$ (resp. $\sigma \in S_\tau$), then there is an edge from τ to σ (resp. from σ to τ).

The conditions 1–3 of T_σ and S_τ translate into the following properties of the graph G:

a. Every vertex has an incoming edge, since the sets T_σ and S_τ are non-empty.
b. Cycles in G have length at most 2. Indeed, by property 2, an edge from τ to σ implies that the size

of the antecedent of τ is no less than the size of σ. But then all s-t tgds associated to the vertices in a cycle must have antecedents of equal size. By properties 1 and 2, all such antecedents are isomorphic. This means that the conclusions are isomorphic as well, since they are obtained as cores of the chase of isomorphic source instances with equivalent sets of dependencies (procedure PROPAGATE).

c. Vertices that participate in such a two-edge cycle are disconnected from the rest of the graph. This follows from the fact that the corresponding s-t tgds are equivalent, and thus any other edge would contradict the property 3.

We obtained a graph, of which each vertex should be connected by an incoming path to a cycle (there is only a finite number of vertices, and from each vertex an infinite incoming path can be traced, by the property "a"). Considering "c", this is only possible if each vertex itself belongs to a cycle, and, by "b", G must consist of connected components of size 2. In total, this means, that every s-t tgd $\sigma \in {\Sigma_{st}}^*$ has an isomorphic counterpart $\tau \in \Upsilon^*_{st}$ and vice versa. □

The question now is how to further simplify the set of s-t tgds. Due to the egds, we could strengthen Rule 5 from Section 3 (i.e., deletion of redundant atoms from some conclusion) to the Rule E3 in Figure 4. Unfortunately, this would again lead to a non-unique normal form as the following example illustrates.

Example 15 Consider the mapping consisting of two s-t tgds and one egd:

$S(x,y) \to P(x,z) \wedge Q(x,z)$
$S(x,y) \to R(x,z) \wedge Q(x,z)$
$P(x,z_1) \wedge R(x,z_2) \to z_1 = z_2$

It is easy to verify that the atom Q can be eliminated by the rule E3 from the conclusion of any of the two tgds, but not from both. □

However, if we content ourselves with the simplifications from the s-t tgds only case (i.e,. Rules 1 – 5 from Section 3), then we get an intuitive normal form which is simplified to a large extent and which is guaranteed to be unique up to isomorphism. As was mentioned earlier, it is sometimes important in data exchange to arrive at a unique canonical universal solution (this is in particular the case for defining the semantics of queries in a way that the semantics does not not depend on the syntax of the dependencies). In these situations, the normal form defined below should be chosen.

Definition 12 Consider a set $\Sigma = \Sigma_{st} \cup \Sigma_t$ of s-t tgds Σ_{st} and target egds Σ_t and let the result of PROPAGATE(Σ_{st}, Σ_t) be denoted by $(\Sigma_s, {\Sigma_{st}}')$ Moreover, let ${\Sigma_{st}}^*$ denote the set of s-t tgds resulting from ${\Sigma_{st}}'$ by exhaustive application of the rules E1, E2 as well as the rules 1–5 from Section 3 and let Σ^*_s denote the result of exhaustive reduction of Σ_s via rule E1. Then we call $\langle \Sigma^*_s, {\Sigma_{st}}^*, \Sigma_t \rangle$ the *normal form* of Σ.

Theorem 7 *Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ and $\Upsilon = \Upsilon_{st} \cup \Upsilon_t$ be equivalent sets consisting of s-t tgds and target egds and let $\langle \Sigma^*_s, {\Sigma_{st}}^*, \Sigma_t \rangle$ and $\langle \Upsilon^*_s, \Upsilon^*_{st}, \Upsilon_t \rangle$ be the corresponding normal forms. Then ${\Sigma_{st}}^*$ and Υ^*_{st} are isomorphic. Moreover, $\Sigma_t \equiv \Upsilon_t$ holds.*

Proof The fact that ${\Sigma_{st}}^*$ and Υ^*_{st} are isomorphic follows immediately from Theorems 1 and 6. The equivalence $\Sigma_t \equiv \Upsilon_t$ was proved in Lemma 9. □

Homomorphically equivalent components. The normal form obtained by the PROPAGATE procedure followed by the Rules E1 and E2 is not optimal in all respects yet. In particular, both the PROPAGATE procedure and the Rule E2 may have introduced more atoms than needed in the conclusion of s-t tgds. Moreover, by the E2 rule, we may have split the antecedent of tgds into several smaller ones, such that the total number of atoms in the antecedents is increased. Of course, we may now simply apply the rules from Figure 1 to further simplify the set of s-t tgds. However, in the final part of this section, we want to look in a principled way at further optimizations of the normal form of s-t tgds in the presence of egds. The following concept is crucial.

Definition 13 Let $\Sigma = \Sigma_{st} \cup \Sigma_t$. We say that two tgds τ_1 and τ_2 in Σ_{st} are *homomorphically equivalent* if their antecedents are. Moreover, we say two sets S, S' of tgds are homomorphically equivalent if the tgds in one set and the tgds in the other set are homomorphically equivalent.

Obviously, homomorphical equivalence is indeed an equivalence relation on Σ_{st}. We refer to the equivalence classes of this relation as the *HE-components* of Σ_{st}.

We now define a partial order on the HE-components of a set of s-t tgds by considering a "more general" component as greater than a "more specific" one (i.e., there are homomorphisms from the more general one into the "more specific" one but not vice-versa). Moreover, we also consider the closure under the greater-than relation. Below, we show that the closure of each HE-component is unique up to logical equivalence.

Definition 14 Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ be a mapping and let $\mathcal{S} = \{S_1, \ldots, S_m\}$ denote the HE-components of Σ_{st}. We define a *partial order* as follows: for any pair of indices i, j, we define $S_i \geq S_j$ if for every antecedent $\varphi(\mathbf{x})$ of the tgds in S_i and every antecedent $\chi(\mathbf{z})$ of the tgds in S_j, $At(\varphi(\mathbf{x})) \to At(\chi(\mathbf{z}))$ holds (i.e., there is a homomorphism from $\varphi(\mathbf{x})$ to $\chi(\mathbf{z})$). If $S_j \not\geq S_i$, S_i is said to be strictly greater than S_j, $S_i > S_j$.

For $i \in \{1, \ldots, n\}$, we define the *closure of S_i above* as $Cl_{\geq}(S_i, \Sigma) = \{\tau \mid \tau \in S_j$ for some j with $S_j \geq S_i\}$.

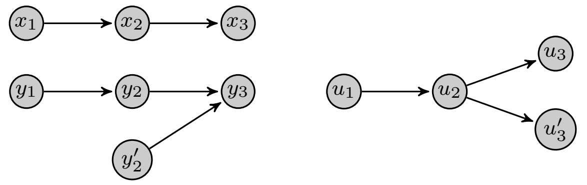

Fig. 5 Antecedents of τ_1 (left) and τ_2 (right), Example 16

Example 16 Consider a source schema consisting of a single relation symbol $P(\cdot,\cdot)$ and a schema mapping $\Sigma = \{\tau_1, \tau_2, \tau_3, \tau_4\}$, where the τ_i's are defined as follows:

$$\begin{aligned}
&\tau_1 : P(x_1, x_2) \wedge P(x_2, x_3) \wedge \\
&\qquad P(y_1, y_2) \wedge P(y_2, y_3) \wedge P(y_2', y_3) \to Q(x_1, y_3) \\
&\tau_2 : P(u_1, u_2) \wedge P(u_2, u_3) \wedge P(u_2, u_3') \to Q(u_3, u_3') \\
&\tau_3 : P(v_1, v_2) \to T(v_1, v_2) \\
&\tau_4 : P(v_1, v_1) \to Q(v_1, v_1)
\end{aligned}$$

Intuitively, the binary relation symbol $P(\cdot,\cdot)$ can be thought of as defining edges of a directed graph. Then the antecedent of the tgd τ_1 consists of two connected components: two paths of length two, one having an additional edge pointing to the peak. The antecedent of the tgd τ_2 corresponds to a Y-shaped graph (see Figure 5). The antecedent of τ_3 consists of a single edge, and the antecedent of τ_4 consists of a single self-loop.

The antecedents of τ_1 and τ_2 have the same cores (a path of length 2) and thus are homomorphically equivalent. Hence, τ_1 and τ_2 are part of the same HE-component S_1. The tgd τ_3 belongs to a different HE-component S_2 with $S_2 > S_1$. Indeed, there is a homomorphism sending $P(v_1, v_2)$ either to the antecedent of τ_1 or the antecedent of τ_2, but not vice versa. For the same reason, τ_4 gives rise to yet another HE-component S_3 with $S_1 > S_3$. In total, Σ has three HE-components. As far as the "closure above" is concerned, we thus have $Cl_{\geq}(S_1, \Sigma) = \{\tau_1, \tau_2, \tau_3\}$, $Cl_{\geq}(S_2, \Sigma) = \{\tau_3\}$, and $Cl_{\geq}(S_3, \Sigma) = \Sigma$.

Lemma 17 *Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ and $\Upsilon = \Upsilon_{st} \cup \Upsilon_t$ be two logically equivalent mappings. Moreover, let S be an HE-component in Σ_{st} and let T be an HE-component in Υ_{st}, s.t. S and T are homomorphically equivalent. Then $Cl_{\geq}(S, \Sigma) \cup \Sigma_t \equiv Cl_{\geq}(T, \Upsilon) \cup \Upsilon_t$ holds.*

Proof By Lemma 9 we have $\Sigma_t \equiv \Upsilon_t$. It remains to show that, for every $\tau \in Cl_{\geq}(T, \Upsilon)$, the implication $Cl_{\geq}(S, \Sigma) \cup \Sigma_t \models \tau$ holds. The implication $Cl_{\geq}(T, \Upsilon_{st}) \cup \Upsilon_t \models \sigma$ for every $\sigma \in S$ follows by symmetry.

By $\Sigma \equiv \Upsilon$, we clearly have $\Sigma \models \tau$. Let $\varphi(\mathbf{x})$ denote the frozen antecedent of τ and let $I = At(\varphi(\mathbf{x}))$. Now consider the result $I^{\Sigma_{st}}$ of chasing I with Σ_{st}: Clearly, only those tgds $\sigma \in \Sigma_{st}$ fire, s.t. there is a homomorphism from the antecedent of σ to $\varphi(\mathbf{x})$. These are precisely the tgds in $Cl_{\geq}(S, \Sigma_{st})$. Hence, we have $I^{\Sigma_{st}} = I^{Cl_{\geq}(S, \Sigma)}$. But then, by the implication criterion of [4] recalled in Lemma 10, $\Sigma \models \tau$ holds iff $Cl_{\geq}(S, \Sigma) \cup \Sigma_t \models \tau$ holds. □

The following lemma shows that, unless a mapping contains redundant dependencies, the HE-components of a mapping are in a sense invariant under logical equivalence. Moreover, HE-components may be exchanged between logically equivalent mappings.

Lemma 18 *Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ and $\Upsilon = \Upsilon_{st} \cup \Upsilon_t$ be two logically equivalent mappings, s.t. Rule E1 is not applicable to them. Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ denote the HE-components of Σ_{st} and $\mathcal{T} = \{T_1, \ldots, T_n\}$ the HE-components of Υ_{st}. Then the following properties hold: $n = m$ and for every $S_i \in \mathcal{S}$, there exists exactly one j, s.t. the tgds in S_i are homomorphically equivalent to the tgds in T_j.*

Proof W.l.o.g. suppose that there exists an HE-component S_i of Σ_{st} which is not homomorphically equivalent to any HE-component T_j of Υ_{st}. By assumption, $\Sigma \equiv \Upsilon$. Hence, $\Upsilon \models S_i$. Let $\mathcal{T}^* \subseteq \mathcal{T}$ with $\mathcal{T}^* = \bigcup\{T_j \mid T_j \geq S_i\}$. By the same considerations as in the proof of Lemma 17, only the HE-components in $\mathcal{T}^*$ are used to test the implication $\Upsilon \models S_i$ via Lemma 10. Hence, we have $\mathcal{T}^* \models S_i$.

On the other hand, also $\Sigma \models \mathcal{T}^*$. Now define $\mathcal{S}^* = \bigcup\{S_k \mid S_k \geq T_j \text{ for some } T_j \in \mathcal{T}^*\}$. Again, we may conclude $\mathcal{S}^* \models \mathcal{T}^*$ and, therefore, also $\mathcal{S}^* \models S_i$ By assumption, Υ does not contain an HE-component whose tgds are homomorphically equivalent to S_i. Therefore, all HE-components in $\mathcal{T}^*$ are strictly greater than S_i. But then, all HE-components S_k in $\mathcal{S}^*$ are also strictly greater than S_i. Thus, $\Sigma \setminus S_i \models S_i$. In other words, every dependency in S_i can be removed from Σ_{st} by the Rule E1, which contradicts the assumption that the E1 Rule is not applicable. □

Lemma 19 *Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ and $\Upsilon = \Upsilon_{st} \cup \Upsilon_t$ be two logically equivalent mappings, s.t. the E1 Rule is not applicable to them. Moreover, let S be an HE-component in Σ_{st} and let T be an HE-component in Υ_{st}, s.t. S and T are homomorphically equivalent. Then the logical equivalence $\Sigma \equiv (\Sigma_{st} \setminus S) \cup T \cup \Sigma_t$ holds (i.e., we may replace the HE-component S from Σ by the corresponding HE-component T from Υ).*

Proof Let $\mathcal{S} = \{S_1, \ldots, S_n\}$ denote the HE-components of Σ_{st} and $\mathcal{T} = \{T_1, \ldots, T_n\}$ the HE-components of Υ_{st}. By Lemma 18, we may assume w.l.o.g., that every S_i is homomorphically equivalent to T_i. Now let S and T of this lemma correspond to S_j and T_j, for some $j \in \{1 \ldots n\}$.

We apply Lemma 17 to all HE-components that are strictly greater than S_j resp. T_j: Let $I = \{i \mid S_i > S_j\}$. Clearly, $I = \{i \mid T_i > T_j\}$. For every $i \in I$, we have $Cl_{\geq}(S_i, \Sigma) \cup \Sigma_t \equiv Cl_{\geq}(T_i, \Upsilon) \cup \Upsilon_t$ by Lemma 17. Then also $(\bigcup_{i \in I} Cl_{\geq}(S_i, \Sigma)) \cup \Sigma_t \equiv (\bigcup_{i \in I} Cl_{\geq}(T_i, \Sigma)) \cup \Upsilon_t$ holds, i.e.: $(Cl_{\geq}(S_j, \Sigma) \setminus S_j) \cup \Sigma_t \equiv (Cl_{\geq}(T_j, \Sigma) \setminus T_j) \cup \Upsilon_t$, i.e., the HE-components *strictly* greater than S_j and T_j lead to logical equivalence.

Now if we apply Lemma 17 to S_j and T_j, we may conclude $Cl_{\geq}(S_j, \Sigma) \cup \Sigma_t \equiv Cl_{\geq}(T_j, \Upsilon) \cup \Upsilon_t$. By the above considerations, we may exchange in $Cl_{\geq}(T_j, \Upsilon)$ all HE-components that are *strictly* greater than T_j by the corresponding HE-components from Σ. That is, $Cl_{\geq}(S_j, \Sigma) \cup \Sigma_t \equiv (Cl_{\geq}(S_j, \Sigma) \setminus S_j) \cup T_j \cup \Upsilon_t$. By adding all remaining HE-components of Σ to both sides of the equivalence, we get the desired equivalence $\Sigma \equiv (\Sigma_{st} \setminus S) \cup T \cup \Sigma_t$. □

HE-components will turn out to be crucial for optimizing the s-t tgds. Indeed we show that for all optimization criteria considered here, local optimization inside every HE-component yields a global optimum.

Definition 15 An optimization problem on sets of dependencies is called a *sum-minimization problem* if the goal of the optimization is to minimize a function F with the following property: (1) $F(\Sigma) \geq 0$ holds for every set of dependencies Σ and (2) for any two sets of dependencies Σ, Σ' with $\Sigma \cap \Sigma' = \emptyset$, we have $F(\Sigma \cup \Sigma') = F(\Sigma) + F(\Sigma')$.

Clearly, all optimization criteria studied here (like cardinality-minimality, antecedent-minimality, conclusion-minimality, and variable-minimality, see Definition 2) are sum-minimization problems.

Definition 16 Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ be a mapping, s.t. the E1 Rule is not applicable to it, i.e., Σ contains no s-t tgd that may be deleted. Now consider a sum-minimization problem whose goal is to minimize some function F over sets of s-t tgds.

We say that Σ is *globally optimal* (or simply *optimal*) if, for every mapping $\Upsilon = \Upsilon_{st} \cup \Upsilon_t$ with $\Sigma \equiv \Upsilon$, we have $F(\Sigma) \leq F(\Upsilon)$.

We say that Σ is *locally optimal* if the following conditions are fulfilled: let $\Upsilon = \Upsilon_{st} \cup \Upsilon_t$ be an arbitrary mapping with $\Sigma \equiv \Upsilon$. Moreover, let S be an arbitrary HE-component of Σ and let T be the corresponding HE-component of Υ_{st}, s.t. S and T are homomorphically equivalent. Then $F(S) \leq F(T)$ holds.

Theorem 8 *Let $\Sigma = \Sigma_{st} \cup \Sigma_t$ be a mapping, s.t. the E1 Rule is not applicable to it. Now consider a sum-minimization problem whose goal is to minimize some function F over sets of s-t tgds. Then Σ is* globally optimal *iff it is* locally optimal.

Proof Let $\Upsilon = \Upsilon_{st} \cup \Upsilon_t$ be an arbitrary mapping with $\Sigma \equiv \Upsilon$. By Lemma 18, there exist sets of s-t tgds $\mathcal{S} = \{S_1, \dots, S_n\}$ and $\mathcal{T} = \{T_1, \dots, T_n\}$, s.t. $\mathcal{S}$ denotes the set of HE-components of Σ_{st}, $\mathcal{T}$ denotes the set of HE-components of Υ_{st}, and for every i, the tgds in S_i are homomorphically equivalent to the tgds in T_i.

First suppose that Σ is *globally optimal*. We have to show that then Σ is also locally optimal. Assume to the contrary that $F(S_i) > F(T_i)$ holds for some i. We define $\Sigma' = (\Sigma_{st} \setminus S_i) \cup T_i \cup \Sigma_t$. By Lemma 19, $\Sigma \equiv \Sigma'$. Moreover, since we are considering a sum-minimization problem, we clearly have: $F(\Sigma_{st}) = F(\Sigma_{st} \setminus S_i) + F(S_i) > F(\Sigma_{st} \setminus S_i) + F(T_i) = F((\Sigma_{st} \setminus S_i) \cup T_i) = F(\Sigma')$. This contradicts the assumption that Σ is globally optimal.

Now suppose that Σ is *locally optimal*. We have to show that then Σ is also globally optimal The local optimality implies that $F(S_i) \leq F(T_i)$ holds for every i. Since F defines a sum-minimization problem, we have $F(\Sigma_{st}) = \sum_{i=1}^{n} F(S_i)$ and $F(\Upsilon_{st}) = \sum_{i=1}^{n} F(T_i)$. But then also $F(\Sigma_{st}) \leq F(\Upsilon_{st})$ holds, i.e., Σ is globally optimal. □

Theorem 8 says, that for the optimization of an HE-component, it does not matter how and if other HE-components have already been optimized. However, this does not mean that one can optimize a single HE-component in isolation. In particular, the closure above must be considered.

As demonstrated by the Examples 14 and 15, aggressive splitting and conclusion optimization lead to a non-unique normal form. In the rest of the section, we consider an operation opposite to splitting: Namely, merging of multiple s-t tgds, to enforce cardinality-minimality. As we will see, also this approach leads to non-unique normal forms. The following theorem contains a property that any merge operation must fulfill:

Theorem 9 *Consider a mapping $\Sigma = \Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ created by the* Propagate *procedure, and additionally reduced by the rules E1 and E2. Assume that dependency $\tau \in \Sigma_{st}$ with the antecedent φ can be substituted by the dependency τ' with the antecedent φ', such that $\Sigma \cup \{\tau'\} \setminus \{\tau\} \equiv \Sigma$ holds, and $At(\varphi')$ does not cause a chase failure under Σ. Then, φ must coincide (up to isomorphism) with some endomorphic image of φ'.*

Proof By Theorem 6, exhaustive application of the rules E1 and E2 allows us to obtain a unique normal form of s-t dependencies. Hence, if the mapping $\Sigma' = \Sigma \cup \{\tau'\} \setminus \tau$ is logically equivalent to Σ, it is possible to bring it back in the form isomorphic to Σ by applying the procedure Propagate, followed by the rules E1 and E2.

Since $At(\varphi')$ does not cause a chase failure, we know that Propagate does not affect φ' in any way. Moreover, all the remaining rules in $\Sigma \setminus \{\tau\}$ remain unchanged after Σ' is transformed by E1 and E2. Hence, it must be the case that one can obtain τ from τ' by (possibly successive) applications of E2, and hence φ has to be among the endomorphic images of φ'. □

To achieve cardinality-minimality, we will replace each HE-component with a single tgd. As Theorem 9 suggests, the antecedent of this tgd must contain every antecedent from the original HE-component as an endomorphic image. The following example illustrates that there is no unique minimal “merged” antecedent.

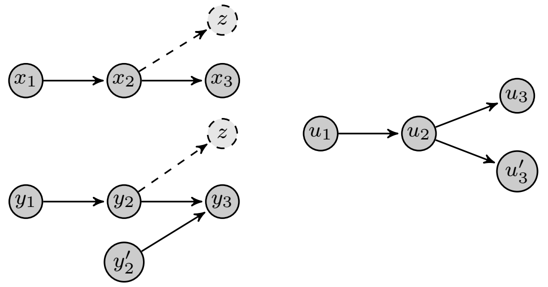

Fig. 6 Possible merges of antecedents φ_1, φ_2, Example 17

Example 17 Recall the mapping Σ from Example 16, with the HE-component S_1 containing tgds τ_1, τ_2:

$$\tau_1 : P(x_1, x_2) \wedge P(x_2, x_3) \wedge \\ P(y_1, y_2) \wedge P(y_2, y_3) \wedge P(y_2', y_3) \rightarrow Q(x_1, y_3)$$

$$\tau_2 : P(u_1, u_2) \wedge P(u_2, u_3) \wedge P(u_2, u_3') \rightarrow Q(u_3, u_3')$$

Let φ_1, φ_2 denote the antecedents of τ_1 and τ_2, respectively. Recall the graphical representation of φ_1 and φ_2 that was given in Figure 5. Obviously, φ_1 and φ_2 are not isomorphic.

Now, there are two ways of adding a single edge to φ_1 in order to get a minimum conjunctive query containing both antecedents as its endomorphic images namely, $\varphi_1' = \varphi_1 \wedge P(x_2, z)$ and $\varphi_1'' = \varphi_1 \wedge P(y_2, z)$, see Figure 6. Clearly, the resulting antecedents φ_1' and φ_1'' are not isomorphic. □

Example 17 shows that there is no unique optimal way of merging s-t tgds from a single HE-component. Notably, egds play no role here. On the other hand, an obvious unique (though hardly optimal) way of merging would be to take a conjunction of all antecedents in a HE-component of a mapping resulting from the exhaustive application of the rules E1 and E2, and renaming apart the variables in distinct tgds.

We conclude this discussion by presenting a procedure that merges several homomorphically equivalent conjunctive queries in one, of reasonable size and satisfying the condition of Theorem 9. At every iteration, the procedure Merge takes two conjunctive queries φ_i and φ_j, finds a greatest common (up to isomorphism) endomorphic image in them, which in the worst case is the core, and renames the variables of φ_i in such a way that it is stitched to φ_j along this greatest common endomorphic subquery. The resulting query φ_{ij} is thus sure to have an endomorphism to φ_i as well as to φ_j.

This operation is then used in the Procedure MergeTgds, which produces a s-t tgd to substitute a given HE-component in a mapping Σ. To build the conclusion of such a merged tgd, MergeTgds uses the Propagate procedure, which chases the merged antecedent with Σ and then takes the conjunction of atoms in the core of the resulting target instance as the conclusion.

```
Procedure Merge
Input. A set $\Phi$ of homomorphically equivalent CQs;
Output. CQ $\varphi$ having endomorphism onto each $\varphi_i \in \Phi$.
  while $|\Phi| > 1$ do
    Choose distinct $\varphi_i, \varphi_j \in \Phi$.
    Find an endomorphism $e_i$ for $\varphi_i$ and $e_j$ for $\varphi_j$, such that
        $e_i(\varphi_i) \cong e_j(\varphi_j)$ and $|e_i(\varphi_i)|$ is maximized.
    Let $\lambda$ be a variable renaming $e_i(\varphi_i) \rightarrow e_j(\varphi_j)$.
    Set $\varphi_{ij} := \varphi_i\lambda \wedge \varphi_j$.
    Set $\Phi := \Phi \cup \{\varphi_{ij}\} \setminus \{\varphi_i, \varphi_j\}$.
  od;
  return the element remaining in $\Phi$;

Procedure MergeTgds
Input. A mapping $\Sigma = \Sigma_s \cup \Sigma_{st} \cup \Sigma_t$, a CQ $\chi$;
Output. A tgd $\tau$ and set $\Sigma_s'$ of source egds, such that
        the equality $\Sigma \equiv (\Sigma \setminus \Sigma[\chi]) \cup \Sigma_s' \cup \{\tau\}$ holds.
/*(a) collect and merge the antecedents of $\Sigma[\chi]$ */
   Set $\Phi = \{\varphi_i | (\varphi_i(\mathbf{x}_i) \rightarrow (\exists \mathbf{y}_i)\,\psi(\mathbf{x}_i, \mathbf{y}_i) \in \Sigma_{st}) \wedge \varphi_i \leftrightarrow \chi\}$;
   $I := At(\text{Merge } (\Phi))^{\Sigma_s}$
/*(b) initialize $\tau'$ */
  $J := I^{\Sigma_{st}}$.
  Let $\mathbf{y}$ be a tuple of all labeled nulls from $var(J) \setminus var(I)$
  $\tau' := \bigwedge_{A \in I} A \rightarrow (\exists \mathbf{y}) \bigwedge_{B \in J} B$.
/*(c) propagate $\Sigma_t$ through $\tau'$ */
  $(\Sigma_s', \Sigma_{st}^*) :=$ Propagate $(\Sigma \cup \{\tau'\})$.
/*(d) return the result */
  Let $\tau$ be the s-t tgd in $\Sigma_{st}^*$ corresponding to $\tau'$.
  return $(\tau,\ \Sigma_s')$;
```

Fig. 7 Procedures Merge and MergeTgds.

Definition 17 Let Σ be a set of s-t tgds and let χ be a conjunctive query. Then we write $\Sigma[\chi]$ to denote the HE-component of those tgds in Σ, whose antecedents are homomorphically equivalent to χ.

Theorem 10 *Let $\Sigma = \Sigma_s \cup \Sigma_{st} \cup \Sigma_t$ be a mapping consisting of source egds, s-t tgds and target egds, Σ_s and Σ_{st} being produced by the* Propagate *procedure, and let χ be a CQ. Moreover, let (τ, Σ_s') be the output of* MergeTgds (Σ, χ)*. Then, the following equivalence holds: $\Sigma \equiv (\Sigma \setminus \Sigma[\chi]) \cup \Sigma_s' \cup \{\tau\}$.*

Proof First, the new s-t tgd τ' was created in step (b) of MergeTgds by chasing with Σ, so $\Sigma \models \tau'$ holds and thus $\Sigma \equiv \Sigma \cup \{\tau'\}$. But then, also $\Sigma \models \Sigma \cup \Sigma_s' \cup \{\tau\}$ follows by Lemma 12, as both Σ_s' and τ were produced by applying Propagate Procedure to $\Sigma \cup \{\tau'\} \equiv \Sigma$.

In the other direction, the merged antecedent $\varphi(\mathbf{x})$ is at least as "powerful" as any of the antecedent $\varphi_i(\mathbf{x}_i)$ in $\Sigma[\chi]$, in the following sense: whenever a substitution λ for the variables $\mathbf{x}_i$ exists, such that $At(\varphi_i(\mathbf{x}_i\lambda)) = I$, then also for some substitution μ for $\mathbf{x}$, $At(\varphi(\mathbf{x}\mu)) = I$.

Indeed, suppose that at step (a) of MergeTgds, the property Merge $(\Phi) \models \Sigma$ holds. Then, the claim is immediate, since Merge is designed to deliver a CQ that satisfies Theorem 9. If, however, Merge (Φ) has to be updated with Σ_s, the unifications performed by Σ_s do not affect the property that every CQ φ_i in Φ is an endomorphic image of φ. Indeed, let μ be an endomorphism of $\varphi(\mathbf{x}) = \textsc{Merge}\ (\Phi)$ transforming it into some $\varphi_i \in \Phi$. For every such μ, we have $At(\varphi(\mathbf{x}\mu)) \models \Sigma_s$, so whenever two nulls $v, w \in dom(At(\varphi))$ are to be unified by Σ_s, necessarily $v\mu = w\mu$ holds. In total, also the instance I created in step (a) of MergeTgds has endomorphisms onto every $At(\varphi_i)$.

Then, also the tgd τ', created in the step (b), is as powerful as τ_i, as for each $\tau_i \in \Sigma_{st}[\chi]$, we know that the chase with Σ_{st} has produced at least all the conclusion atoms of τ_i in the conclusion of τ.

The step (c) with Propagate procedure does not affect dependencies other than τ', since, by precondition of the theorem, Σ_{st} results from Propagate procedure, and thus, no frozen antecedent database of Σ_{st} can cause chase failure under Σ. Moreover, an application of Propagate to τ' cannot deteriorate any endomorphism, which makes the antecedent $\varphi'(\mathbf{x}')$ of τ' isomorphic to some $\varphi_i \in \Phi$. Indeed, suppose this happens and the unification of variables $x'_k, x'_l \in \mathbf{x}'$ cancels some endomorphism λ. That is, λ is such that $\varphi'(\mathbf{x}'\lambda) = \varphi_i(\mathbf{x}_i)$, and $x'_k\lambda \neq x'_l\lambda$. Then, in step 2.(d) of the Propagate procedure, the source egd ε: $\varphi'(\mathbf{x}') \rightarrow x'_k = x'_l$ is produced, and $At(\varphi_i) \not\models \varepsilon$ must be the case. Hence, by $\Sigma \equiv \Sigma \cup \{\tau'\}$ and Claim 2 of Lemma 11, the chase of frozen $At(\varphi_i)$ with Σ fails, which contradicts Claim 1 of Lemma 11 and the fact that Σ_{st} is the output of Propagate. □

Example 18 Recall the tgds τ_1 and τ_2 with the antecedents φ_1 and φ_2 from Example 17. As illustrated by that example, there are two possible ways to merge φ_1 and φ_2, resulting in two possible merged antecedents $\varphi'_1 = \varphi_1 \wedge P(x_2, z)$ and $\varphi''_1 = \varphi_1 \wedge P(y_2, z)$. The corresponding outputs of the procedure MergeTgds are

$$\tau'_1 : P(y_1, y_2) \wedge P(y_2, y_3) \wedge P(y'_2, y_3) \wedge P(x_1, x_2) \wedge P(x_2, x_3) \wedge P(x_2, z) \rightarrow Q(x_1, y_3) \wedge Q(x_3, z)$$

and

$$\tau''_1 : P(y_1, y_2) \wedge P(y_2, y_3) \wedge P(y'_2, y_3) \wedge P(y_2, z) \wedge P(x_1, x_2) \wedge P(x_2, x_3) \rightarrow Q(x_1, y_3) \wedge Q(y_3, z),$$

respectively. □

Summary. To sum up, the following lessons have been learned from our analysis of the normalization and optimization of s-t tgds in the presence of egds: In contrast to the tgd-only case, we have seen that one has to be very careful with the definition of splitting and optimization so as not to produce a non-unique normal form: If we aim at a strict generalization of the splitting rule from Section 3 via the Rule ES in Figure 4, then there does not exist a unique normal form. This also happens if we aim at a strict generalization of the Rule 5 (deletion of redundant atoms from the conclusion of a tgd) via the Rule E3 in Figure 4. For most purposes, we therefore consider the transformation of an arbitrary mapping (consisting of s-t tgds and target egds) via the Propagate procedure and exhaustive application of the rules E1 and E2 from Figure 4 followed by the Rules 1 – 5 from Section 3 as the best choice: The resulting normal form is unique up to isomorphism and incorporates a reasonable amount of splitting and simplification. From the splitting point of view, the resulting normal form is referred to as "antecedent-split-reduced". This corresponds to a restriction of the splitting rule in the tgd-only case to those situations where subsequent antecedent simplifications of all resulting dependencies are possible. Such a restriction is justifiable by the fact that one of the main motivations for splitting is indeed to further reduce the antecedents. From the optimization point of view, the Rules 1–5 guarantee that we do not perform worse than in the tgd-only case. But of course, this leaves some additional potential of further optimization in the presence of egds (in particular the Rule E3) unexploited.

We have also identified the HE-components (components of tgds with homomorphically equivalent antecedents) as an important handle for the most common optimization tasks on the s-t tgds (in particular, for all optimization criteria according to Definition 2). We have seen that a global optimum according to the optimization criteria studied here is obtained by locally optimizing the s-t tgds inside each HE-component. In particular, this allowed us to define a simple procedure which transforms a mapping into an equivalent one with the smallest possible number of s-t tgds. Of course, also in this case, uniqueness is not guaranteed.

We have entirely concentrated on the normalization and optimization of the s-t tgds, while a transformation of the egds has not been considered. Indeed, a normal form of the (source or target) egds is not important for our purposes since we will show in Theorem 11 that the unique (up to isomorphism) canonical universal solution in data exchange only depends on the normalization of the s-t tgds – the equivalence of the source egds and the concrete syntax of the egds are irrelevant.

5 Aggregate Queries

As an application for the schema mappings normalization, in this chapter we discuss the semantics and evaluation of aggregate queries in data exchange, i.e., queries of the form SELECT f FROM R, where f is an aggregate operator min($R.A$), max($R.A$), count($R.A$), count($*$), sum($R.A$), or avg($R.A$), and where R is a target relation symbol or, more generally, a conjunctive query over the target schema and A is an attribute of R. For this purpose, we first recall some basic notions on query answering in data exchange as well as some fundamental results on aggregate queries from [1].

Certain Answers. Though any target database satisfying the schema mapping and local constraints is called a "solution", a random choice of a candidate for materializing a target database is not satisfactory: query answering in data exchange cannot be reduced to evaluating queries against random solutions. The widely accepted approach is based on the notion of *certain answers*:

Definition 18 Let Σ be a schema mapping over the schema $\langle \mathbf{S}, \mathbf{T} \rangle$, and let I be an instance over $\mathbf{S}$. Then, the certain answer for a query q over $\mathbf{T}$ and for I is $\mathit{certain}(q, I, \mathcal{W}(I)) = \bigcap\{q(J) | J \in \mathcal{W}(I)\}$, where $\mathcal{W}(I)$ is the *set of possible worlds* for I and Σ.

Several proposals can be found in the literature [9,14, 19,20] as to which solutions should be taken as possible worlds $\mathcal{W}(I)$. Typical examples are the set of all solutions, the set of universal solutions, the core of the universal solutions, or the CWA-solutions. For conjunctive queries, all these proposals lead to identical results.

Aggregate Certain Answers. Afrati and Kolaitis [1] initiated the study of the semantics of aggregate queries in data exchange. They adopted the notion of *aggregate certain answers* for inconsistent databases of Arenas et al. [3] to data exchange:

Definition 19 [1] Let query q be of the form `SELECT` f `FROM` R, where R is a target relation symbol or, more generally, a first-order query over the target schema $\mathbf{T}$, and f is one of the following aggregate operators: $\mathsf{min}(R.A)$, $\mathsf{max}(R.A)$, $\mathsf{count}(R.A)$, $\mathsf{count}(*)$, $\mathsf{sum}(R.A)$, or $\mathsf{avg}(R.A)$ for some attribute A of R. For all aggregate operators but $\mathsf{count}(*)$, tuples with a null value in attribute $R.A$ are ignored in the computation.

- Value r is a possible answer of q w.r.t. I and $\mathcal{W}(I)$ if there exists an instance $J \in \mathcal{W}(I)$ for which $f(q)(J) = r$.
- $\mathit{poss}(f(q), I, \mathcal{W}(I))$ denotes the set of all possible answers of the aggregate query $f(q)$ w.r.t. I and $\mathcal{W}(I)$.
- For the aggregate query $f(q)$, the aggregate certain answer $\mathit{agg\text{-}certain}(f, I, \mathcal{W}(I))$ w.r.t. I and $\mathcal{W}(I)$ is the interval
$[\mathsf{glb}(\mathit{poss}(f(q), I, \mathcal{W}(I))),\ \mathsf{lub}(\mathit{poss}(f(q), I, \mathcal{W}(I)))]$, where glb and lub stand, respectively, for the greatest lower bound and the least upper bound.

Semantics of aggregate queries via endomorphic images. A key issue in defining the semantics of queries in data exchange is to define which set of possible worlds should be considered. In [1], Afrati and Kolaitis showed that all previously considered sets of possible worlds yield a trivial semantics of aggregate queries. Therefore, they introduced a new approach via the *endomorphic images of the canonical universal solution*. Let $\mathit{Endom}(I, \mathcal{M})$ denote the endomorphic images of the canonical universal solution $J^* = \mathit{CanSol}(I)$, i.e.: $J \in \mathit{Endom}(I, \mathcal{M})$ if there exists an endomorphism $h \colon J^* \to J^*$, s.t. $J = h(J^*)$. As shown in [1], taking $\mathcal{W}(I) = \mathit{Endom}(I, \mathcal{M})$ leads to an interesting and non-trivial semantics of aggregate queries. However, in general, the semantics definition depends on the concrete syntactic representation of the s-t tgds.

Example 19 Consider the source schema $\mathbf{S} = \{P\}$, target schema $\mathbf{T} = \{R\}$ and the pair of schema mappings $\mathcal{M}_1 = \langle \mathbf{S}, \mathbf{T}, \Sigma_1 \rangle$ and $\mathcal{M}_2 = \langle \mathbf{S}, \mathbf{T}, \Sigma_2 \rangle$ with the following s-t tgds:

$\Sigma_1 = \{P(x) \rightarrow (\exists y) R(1, x, y)\}$ and
$\Sigma_2 = \{P(x) \rightarrow (\exists y_1 \dots y_n) R(1, x, y_1) \wedge \dots \wedge R(1, x, y_n)\}$

Clearly, $\mathcal{M}_1$ and $\mathcal{M}_2$ are logically equivalent. However, for the source instance $I = \{P(a)\}$, they yield different canonical universal solutions $J_1 = \{R(1, a, y)\}$ and $J_2 = \{R(1, a, y_1), \ \dots,\ R(1, a, y_n)\}$. Let A denote the name of the first attribute of R. Then all of the three aggregate queries $\mathsf{count}(R.A)$, $\mathsf{count}(*)$, and $\mathsf{sum}(R.A)$ have the range semantics $[1, 1]$ in $\mathcal{M}_1$ and $[1, n]$ in $\mathcal{M}_2$, i.e.: $\mathcal{M}_1$ admits only one possible world and the three aggregate queries evaluate to 1 in this world. In contrast, $\mathcal{M}_2$ gives rise to a number of possible worlds with $\{R(1, a, y_1), \dots, R(1, a, y_n)\}$ being the biggest one and $\{R(1, a, y)\}$ the smallest. Thus, the three aggregate queries may take values between 1 and n. □

In order to eliminate the dependence on the concrete syntactic representation of the s-t tgds, we have defined a new normal form of s-t tgds in Definition 12. Below, we show that we thus get a unique canonical universal solution also in the presence of target egds.

Theorem 11 *Let $\mathcal{M} = \langle \mathbf{S}, \mathbf{T}, \Sigma_{st} \cup \Sigma_t \rangle$ be a schema mapping and let $\Sigma_s \cup \Sigma^*_{st} \cup \Sigma_t$ be the normal form of $\Sigma_{st} \cup \Sigma_t$. Moreover, let I be a source instance and J^* the canonical universal solution for I under $\mathcal{M}$ obtained via an oblivious chase with Σ^*_{st} followed by a chase with Σ_t in arbitrary order. Then J^* is unique up to isomorphism. We denote J^* as $\mathit{CanSol}^*(I)$.*

Proof (Sketch) By Theorem 6, the normal form of the s-t tgds is unique up to isomorphism. Hence, also the result of the oblivious chase with the s-t tgds is unique up to isomorphism. Finally, also the chase with equivalent sets of egds produces isomorphic canonical universal instances. This property is proved by induction on the length of one of the chase sequences: see Appendix D for details. □

To obtain a unique range semantics of the aggregate functions min, max, count, $\mathsf{count}(*)$, sum, and avg, we therefore propose to follow the approach of [1], with the only difference that we take the unique target instance $\mathit{CanSol}^*(I)$ from Theorem 11.

6 Conclusion

We have initiated the study of a theory of schema mapping optimization. We have thus presented several natural optimality criteria and a rewrite rule system for transforming any set of s-t tgds into an equivalent optimal one. Recently, several other works have also presented rewrite rules for transforming a set of s-t tgds into an equivalent one with better computational properties. In [22] and [25], the authors aim at the transformation of a set Σ of s-t tgds into an equivalent set Σ', s.t. chasing a source instance with Σ' directly yields the core of the universal solutions of the corresponding data exchange problem. In [21], this transformation of s-t tgds is extended to mappings which comprise also functional dependencies as target dependencies. The transformations in [22, 25, 21] insert negated atoms and/or inequalities in the antecedents of some s-t tgds so as to block certain forms of applying these s-t tgds in the chase. The goal pursued by these transformations is to avoid the expensive core computation by post-processing of the canonical universal solution and to obtain the core directly as the chase result. Normalization and optimization of the mappings are not in the scope of those transformations.

In order to extend our rewrite rule system to schema mappings including target egds, the most important ingredients of our transformation (namely splitting and simplification of tgds) had to be defined very carefully so as not to destroy the uniqueness of the normal form. We have investigated several forms of splitting and of optimization and we have identified a rewrite rule system which indeed guarantees to produce a normal form that is again unique up to variable renaming. Finally, we have applied the normalization of schema mappings containing target egds to aggregate queries in data exchange. An implementation of the presented algorithms is freely available from `http://www.dbai.tuwien.ac.at/proj/sm`.

In this paper, we have only considerd the optimization of mappings with respect to *logical equivalence*. As pointed out in [10], weaker notions of equivalence such as “data exchange equivalence” and “conjunctive query equivalence” may sometimes be more appropriate. Unfortunately, many optimization tasks in these relaxed settings are undecidable [23].

As future work, our results should be extended to more expressive schema mappings, including second-order s-t tgds or target tgds. Not surprisingly, a unique normal form via redundancy elimination is not feasible for the target tgds: Consider the set of target tgds $\Sigma_t = \{P(x) \to R(x) \land S(x), R(x) \to S(x), S(x) \to R(x)\}$. Now the tgd $P(x) \to R(x) \land S(x)$ can be either reduced to $P(x) \to R(x)$ or to $P(x) \to S(x)$. Of course, even if no unique normal form of the tgds exists, it is still conceivable that one may obtain a unique (up to isomorphism) canonical universal solution via redundancy elimination from the tgds.

References

1. F. N. Afrati and P. G. Kolaitis. Answering aggregate queries in data exchange. In *Proc. PODS'08*, pages 129–138, 2008.
2. M. Arenas, P. Barceló, R. Fagin, and L. Libkin. Locally consistent transformations and query answering in data exchange. In *Proc. PODS'04*, pages 229–240. ACM, 2004.
3. M. Arenas, L. E. Bertossi, J. Chomicki, X. He, V. Raghavan, and J. Spinrad. Scalar aggregation in inconsistent databases. *Theor. Comput. Sci.*, 3(296):405–434, 2003.
4. C. Beeri and M. Y. Vardi. A proof procedure for data dependencies. *J. ACM*, 31(4):718–741, 1984.
5. P. A. Bernstein, T. J. Green, S. Melnik, and A. Nash. Implementing mapping composition. *VLDB J.*, 17(2):333–353, 2008.
6. P. A. Bernstein and S. Melnik. Model management 2.0: manipulating richer mappings. In *Proc. SIGMOD'07*, pages 1–12. ACM, 2007.
7. A. K. Chandra and P. M. Merlin. Optimal implementation of conjunctive queries in relational data bases. In *Proc. STOC'77*, pages 77–90. ACM Press, 1977.
8. R. Fagin. Horn clauses and database dependencies. *J. ACM*, 29(4):952–985, 1982.
9. 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.
10. R. Fagin, P. G. Kolaitis, A. Nash, and L. Popa. Towards a theory of schema-mapping optimization. In *Proc. PODS'08*, pages 33–42. ACM, 2008.
11. R. Fagin, P. G. Kolaitis, and L. Popa. Data exchange: getting to the core. *ACM TODS*, 30(1):174–210, 2005.
12. R. Fagin, P. G. Kolaitis, L. Popa, and W.-C. Tan. Reverse data exchange: coping with nulls. In *Proc. PODS '09*, pages 23–32. ACM, 2009.
13. A. Y. Halevy, A. Rajaraman, and J. J. Ordille. Data integration: The teenage years. In *Proc. VLDB'06*, pages 9–16. ACM, 2006.
14. A. Hernich and N. Schweikardt. Cwa-solutions for data exchange settings with target dependencies. In *Proc. PODS'07*, pages 113–122. ACM, 2007.
15. T. Imielinski and W. L. Jr. Incomplete information in relational databases. *J. ACM*, 31(4):761–791, 1984.
16. D. S. Johnson and A. C. Klug. Testing containment of conjunctive queries under functional and inclusion dependencies. *J. Comput. Syst. Sci.*, 28(1):167–189, 1984.
17. P. G. Kolaitis. Schema mappings, data exchange, and metadata management. In *Proc. PODS'05*, pages 61–75, 2005.
18. M. Lenzerini. Data integration: A theoretical perspective. In *Proc. PODS'02*, pages 233–246. ACM, 2002.
19. L. Libkin. Data exchange and incomplete information. In *Proc. PODS'06*, pages 60–69. ACM Press, 2006.
20. L. Libkin and C. Sirangelo. Data exchange and schema mappings in open and closed worlds. In *Proc. PODS'08*, pages 139–148. ACM, 2008.
21. B. Marnette, G. Mecca, and P. Papotti. Scalable data exchange with functional dependencies. *PVLDB*, 3(1):105–116, 2010.
22. G. Mecca, P. Papotti, and S. Raunich. Core schema mappings. In *Proc. SIGMOD'09*, pages 655–668, 2009.
23. R. Pichler, E. Sallinger, and V. Savenkov. Relaxed notions of schema mapping equivalence revisited. In *Proc. ICDT'11*, pages 90–101, 2011.
24. Y. Sagiv and M. Yannakakis. Equivalences among relational expressions with the union and difference operators. *J. ACM*, 27(4):633–655, 1980.
25. 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.

Appendix

A Proof of Lemma 2

Lemma 2 *The Rules 1 – 5 in Figure 1 are correct, i.e.: Let Σ be a set of s-t tgds and $\tau \in \Sigma$. Suppose that Σ is transformed into Σ' by applying one of the Rules 1 – 5 to τ, that is:*

- *τ is replaced by a single s-t tgd τ' (via Rule 1,2,5),*
- *τ is replaced by s-t tgds $\tau_1, \ldots, \tau_n$ (via Rule 3),*
- *or τ is deleted (via Rule 4).*

Then Σ and Σ' are equivalent.

Proof
Rule 1. Suppose that an s-t tgd τ is replaced by τ' via Rule 1. Then the s-t tgds τ and τ' are of the form τ: $\varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y}\,)\psi(\mathbf{x}, \mathbf{y}\,)$ and τ': $\varphi(\mathbf{x}\,) \rightarrow (\exists \mathbf{y}\,)\psi(\mathbf{x}, \mathbf{y}\sigma)$, s.t. $At(\psi(\mathbf{x}, \mathbf{y}\sigma)) \subset At(\psi(\mathbf{x}, \mathbf{y}))$. In particular, τ' is a "proper instance" of τ according to Definition 7. Hence, by Lemma 3, $\tau' \models \tau$ holds.

On the other hand, let $\langle S, T\rangle$ be an arbitrary pair of source and target instance with $\langle S, T\rangle \models \tau$ and let λ: $\mathbf{x} \rightarrow dom(S)$ be a substitution, s.t. $At(\varphi(\mathbf{x}\,\lambda)) \subseteq S$. We have to show that then also $T \models \psi(\mathbf{x}\lambda, \mathbf{y}\sigma)$ holds. By assumption, $\langle S, T\rangle \models \tau$. Hence, $T \models \psi(\mathbf{x}\lambda, \mathbf{y}\,)$, i.e., there exists a substitution μ, s.t. $At(\psi(\mathbf{x}\lambda, \mathbf{y}\mu)) \subseteq T$. But then, since $At(\psi(\mathbf{x}, \mathbf{y}\sigma)) \subseteq At(\psi(\mathbf{x}, \mathbf{y}\,))$ holds, we also have $At(\psi(\mathbf{x}\lambda, \mathbf{y}\sigma\mu)) \subseteq T$. Thus, $\tau \models \tau'$ indeed holds.

Rule 2. Suppose that τ: $\varphi(\mathbf{x}_1, \mathbf{x}_2) \rightarrow (\exists \mathbf{y}\,)\psi(\mathbf{x}_1, \mathbf{y}\,)$ is replaced by τ' via Rule 2. Then τ' must be of the form τ': $\varphi(\mathbf{x}_1, \mathbf{x}_2\sigma) \rightarrow (\exists \mathbf{y}\,)\psi(\mathbf{x}_1, \mathbf{y}\,)$, with $At(\varphi(\mathbf{x}_1, \mathbf{x}_2\sigma)) \subset At(\varphi(\mathbf{x}_1, \mathbf{x}_2))$. We show both implications $\tau \models \tau'$ and $\tau' \models \tau$ separately.

[$\tau \models \tau'$] Let $\langle S, T\rangle$ be a pair of source and target instance with $\langle S, T\rangle \models \tau$. If $S \not\models \varphi(\mathbf{x}_1, \mathbf{x}_2\sigma)$ then $\langle S, T\rangle \models \tau'$ holds vacuously. It remains to consider the case that $S \models \varphi(\mathbf{x}_1, \mathbf{x}_2\sigma)$ holds, i.e., there exists a substitution λ', s.t. $At(\varphi(\mathbf{x}_1\lambda', \mathbf{x}_2\sigma\lambda')) \subseteq S$. Consider the substitution λ: $\mathbf{x}_1 \cup \mathbf{x}_2 \rightarrow dom(S)$, s.t. $x\lambda = x\lambda'$ for every $x \in \mathbf{x}_1$ and $x\lambda = x\sigma\lambda'$ for every $x \in \mathbf{x}_2$. Then the equality $At(\varphi(\mathbf{x}_1\lambda, \mathbf{x}_2\lambda)) = At(\varphi(\mathbf{x}_1\lambda', \mathbf{x}_2\sigma\lambda')) \subseteq S$ holds. Thus, we conclude that $T \models (\exists \mathbf{y}\,)\psi(\mathbf{x}_1\lambda, \mathbf{y}\,)$, since $\langle S, T\rangle \models \tau$ holds. Hence, since λ and λ' coincide on $\mathbf{x}_1$ also $T \models (\exists \mathbf{y}\,)\psi(\mathbf{x}_1\lambda', \mathbf{y}\,)$ and, therefore, $\langle S, T\rangle \models \tau'$.

[$\tau' \models \tau$] Now let $\langle S, T\rangle$ be a pair of source and target instance with $\langle S, T\rangle \models \tau'$ and $S \models \varphi(\mathbf{x}_1, \mathbf{x}_2)$, i.e., there exists a substitution λ, s.t. $At(\varphi(\mathbf{x}_1\lambda, \mathbf{x}_2\lambda)) \subseteq S$. By definition of Rule 2, the inclusion $At(\varphi(\mathbf{x}_1, \mathbf{x}_2\sigma)) \subseteq At(\varphi(\mathbf{x}_1, \mathbf{x}_2))$ holds. Hence, $At(\varphi(\mathbf{x}_1\lambda, \mathbf{x}_2\sigma\lambda)) \subseteq S$ is true as well. But then, by $\langle S, T\rangle \models \tau'$, we know that also $T \models (\exists \mathbf{y}\,)\psi(\mathbf{x}_1\lambda, \mathbf{y}\,)$ holds and, therefore, $\langle S, T\rangle \models \tau$.

Rule 3. This rule is based on two general equivalences in first-order logic:

- $(\exists \mathbf{z}\,)(A(\mathbf{z}\,) \wedge B(\mathbf{z}\,)) \equiv (\exists \mathbf{z}_1)A(\mathbf{z}_1) \wedge (\exists \mathbf{z}_2)B(\mathbf{z}_2)$ if the variables $\mathbf{z}_1$ actually occurring in A and the variables $\mathbf{z}_2$ actually occurring in B are disjoint.
- We clearly have the equivalence $A \rightarrow (B_1 \wedge B_2) \equiv (A \rightarrow B_1) \wedge (A \rightarrow B_2)$.

Rule 4. Suppose that τ is deleted from Σ via Rule 4, i.e., Σ is transformed into Σ' with $\Sigma' = \Sigma \setminus \{\tau\}$ and $\Sigma' \models \tau$. Hence, $\Sigma' \models \Sigma$ holds. Moreover, $\Sigma \models \Sigma'$ holds by the monotonicity of "$\models$". Hence, $\Sigma \equiv \Sigma'$ clearly holds.
Rule 5. Suppose that $\tau \in \Sigma$ is replaced by τ' via Rule 5. Then τ' is of the form τ': $\varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y}\,')\psi'(\mathbf{x}, \mathbf{y}\,')$, s.t. $At(\psi'(\mathbf{x}, \mathbf{y}\,')) \subset At(\psi(\mathbf{x}, \mathbf{y}\,))$. By the latter condition, $\tau \models \tau'$ clearly holds. Hence, Σ is equivalent to $(\Sigma \cup \{\tau'\})$. By the definition of Rule 5, $(\Sigma \cup \{\tau'\}) \setminus \{\tau\}) \models \tau$ holds. Hence, Σ is also equivalent to $(\Sigma \cup \{\tau'\}) \setminus \{\tau\})$. □

B Full proof of Lemma 6

Lemma 6 *Let τ_1 and τ_2 be two s-t tgds and suppose that τ_1 and τ_2 are reduced w.r.t. Rules 1 – 3. Then τ_1 and τ_2 are isomorphic, iff τ_1 and τ_2 are equivalent.*

Proof The "$\Rightarrow$"-direction is an immediate consequence of Lemma 1. For the "$\Leftarrow$"-direction, consider two equivalent s-t tgds
τ_1: $\varphi_1(\mathbf{x}_1, \mathbf{x}_2) \rightarrow (\exists \mathbf{y}\,)\psi_1(\mathbf{x}_1, \mathbf{y}\,)$ and
τ_2: $\varphi_2(\mathbf{u}_1, \mathbf{u}_2) \rightarrow (\exists \mathbf{v}\,)\psi(\mathbf{u}_1, \mathbf{v}\,)$.

Observe that the antecedents of τ_1 and τ_2 must be homomorphically equivalent, i.e., by Lemma 1, there exist substitutions λ and ρ, s.t.
λ: $\mathbf{x}_1 \cup \mathbf{x}_2 \rightarrow Const \cup \mathbf{u}_1 \cup \mathbf{u}_2$, and
ρ: $\mathbf{u}_1 \cup \mathbf{u}_2 \rightarrow Const \cup \mathbf{x}_1 \cup \mathbf{x}_2$, such that
$At(\varphi_1(\mathbf{x}_1\lambda, \mathbf{x}_2\lambda)) \subseteq At(\varphi_2(\mathbf{u}_1, \mathbf{u}_2))$ and
$At(\varphi_2(\mathbf{u}_1\rho, \mathbf{u}_2\rho)) \subseteq At(\varphi_1(\mathbf{x}_1, \mathbf{x}_2))$.

We show that the antecedents of τ_1 and τ_2 are in fact isomorphic. In particular, we claim that there exist substitutions λ and ρ, s.t. the above inclusions are equalities, i.e.,
λ: $\mathbf{x}_1 \cup \mathbf{x}_2 \rightarrow Const \cup \mathbf{u}_1 \cup \mathbf{u}_2$, and
ρ: $\mathbf{u}_1 \cup \mathbf{u}_2 \rightarrow Const \cup \mathbf{x}_1 \cup \mathbf{x}_2$, such that
$At(\varphi_1(\mathbf{x}_1\lambda, \mathbf{x}_2\lambda)) = At(\varphi_2(\mathbf{u}_1, \mathbf{u}_2))$ and
$At(\varphi_2(\mathbf{u}_1\rho, \mathbf{u}_2\rho)) = At(\varphi_1(\mathbf{x}_1, \mathbf{x}_2))$.

Assume the converse is true, and, w.l.o.g., the antecedent of τ_1 cannot be mapped onto the entire antecedent of τ_2, i.e., for every substitution λ_i: $\mathbf{x}_1 \cup \mathbf{x}_2 \rightarrow Const \cup \mathbf{u}_1 \cup \mathbf{u}_2$, the inclusion $S_{\lambda_i} = At(\varphi_1(\mathbf{x}_1\lambda_i, \mathbf{x}_2\lambda_i)) \subset At(\varphi_2(\mathbf{u}_1, \mathbf{u}_2))$ holds.

For every such λ_i, consider the s-t dependency of the form τ_1^i: $\varphi_1(\mathbf{x}_1\lambda_i, \mathbf{x}_2) \rightarrow (\exists \mathbf{y}\,)\psi_1(\mathbf{x}_1\lambda_i, \mathbf{y}\,)$. Let T denote the complete set of all such τ_1^i. It is easy to see that $\tau_1 \models \tau_2$ iff $T \models \tau_2$. Moreover, by construction of T, $\tau_1 \models T$ and, therefore, we get $\{\tau_1\} \equiv \{\tau_2\} \equiv T$.

Also note that the antecedent database of each $\tau_1^i \in T$ is an endomorphic image of the antecedent database of τ_2. Indeed, assume that for some j, there is no homomorphism projecting $\varphi_2(\mathbf{u}_1, \mathbf{u}_2)$ onto the antecedent $\varphi_1^j(\mathbf{x}_1^j, \mathbf{x}_2^j)$ of $\tau_1^j \in T$. Then, the combined instance $\langle At(\varphi_1^j(\mathbf{x}_1^j, \mathbf{x}_2^j), \emptyset\rangle$ satisfies τ_2 but not T, which is a contradiction.

By Lemma 4 we know that the following two cases are possible: either (i) some $\tau_1^i \models \tau_2$, or (ii) T implies a proper instance of τ_2.

The latter case contradicts Lemma 3, part (2), since we immediately get that τ_2 implies a proper instance of itself. We may therefore assume that (i) is the case: There exists some rule $\tau' \in T$ such that $\tau' \equiv \tau_2$ and moreover, the antecedent of τ' is a proper endomorphic image of the antecedent $\varphi_2(\mathbf{u}_1, \mathbf{u}_2)$ of τ_2.

We also assume that τ' is the smallest "endomorphic" (w.r.t. the antecedent of τ_2) dependency possible. Indeed, let τ' itself admit an equivalent s-t tgd with the proper endomorphically equivalent antecedent: then we just focus on this smaller s-t tgd instead of τ'. Thus, we consider the dependency τ^* whose antecedent $\varphi^*(\mathbf{u}_1^*, \mathbf{u}_2^*)$ which is minimal in the following sense: no dependency logically equivalent to τ_2 (and thus to τ^*) exists, with the antecedent database being a proper endomorphic instance of $At(\varphi^*(\mathbf{u}_1^*, \mathbf{u}_2^*))$.

We show that also the conclusion of τ^* must be smaller than the conclusion of τ_2: that is, the inequality $|At(\varphi_2(\mathbf{u}_1, \mathbf{u}_2))| > |At(\varphi^*(\mathbf{u}_1^*, \mathbf{u}_2^*))|$ holds. Let σ be a substitution, such that $\varphi_2(\mathbf{u}_1\sigma, \mathbf{u}_2\sigma) = \varphi^*(\mathbf{u}_1^*, \mathbf{u}_2^*)$ holds. Since τ^* is minimal, there must also exist a substitution μ: $At(\psi^*(\mathbf{u}_1^*, \mathbf{v}^*\mu)) \subseteq At(\psi(\mathbf{u}_1\sigma, \mathbf{v}))$.

Note that σ necessarily "lumps together" two elements of $\mathbf{u}_1$: otherwise, τ_2 would be not reduced w.r.t. Rule 2 (Core of the antecedent). But then also the inequality $|\psi^*(\mathbf{u}_1^*, \mathbf{v}^*\mu)| < |\psi(\mathbf{u}_1\sigma, \mathbf{v})|$ holds. This means that there exists a dependency equivalent to τ_2 but with fewer conclusion atoms. This contradicts the assumption that $\{\tau_2\}$ is reduced w.r.t. Rule 5.

That is, we have shown that the antecedents of τ_2 and τ_1 are isomorphic. But since these dependencies are equivalent and reduced w.r.t. Rule 1 (Core of the conclusion), we also have that the entire τ_1 and τ_2 must be isomorphic as well. □

C Full proof of Theorem 2

Theorem 2 *Suppose that the length (i.e., the number of atoms) of the s-t tgds under consideration is bounded by some constant b. Then there exists an algorithm which reduces an arbitrary set Σ of s-t tgds to normal form in polynomial time w.r.t. the total size $||\Sigma||$ of (an appropriate representation of) Σ.*

Proof Let constant b limit the number of atoms in each s-t tgd and let $||\Sigma|| = n$ denote the number of s-t tgds in Σ. Moreover, let α denote the maximum arity of the relation symbols in the source and target schema. We define the following simple algorithm:

1. Obtain Σ' by applying Rules 1 – 3 exhaustively to $\{\tau\}$, for each $\tau \in \Sigma$.
2. For each τ in the current set Σ' of s-t tgds do
   - Try to delete τ via Rule 4.
   - If τ was not deleted, try to replace τ by some τ' via Rule 5.
   - If Rule 5 was applicable, apply Rule 3 to τ'.

Note that the Rules 1 and 2 do not become applicable anymore after an application of Rule 5 provided that we replace τ by an s-t tgd τ' such that the set of atoms in the conclusion of τ' is minimal.

In order to establish the polynomial-time upper bound, we proceed in 2 steps. That is, we prove (1) an upper bound on the total number of rule applications and (2) an upper bound on the cost of each single rule application.

(1) *Total number of rule applications.* Rule 4 deletes an s-t tgd. Hence, it can be applied at most n times. The Rules 1, 2, and 5 delete at least one atom from an s-t tgd. Hence, in total, these rules can be applied at most $b*n$ times. Finally, Rule 3 splits the conclusion of an s-t tgd in 2 or more parts. Hence, also the total number of applications of Rule 3 is bounded by $b*n$. We thus get the upper bound $O(b*n)$ on the total number of applications of any rule. Moreover, it should be noted that at no stage of the algorithm, the current set Σ' of s-t tgds contains more than $b*n$ s-t tgds.

(2) *Cost of a single rule application.* In Rules 1 and 2, we compute the core of the atoms in the conclusion resp. in the antecedent. Rules 1 and 2 thus essentially come down to CQ answering of a query with $\leq b$ atoms over a database with $\leq b$ atoms. The cost of a single application of these rules is therefore in $O(\alpha b^b)$.

Rule 3 is the cheapest one in that it only requires the computation of the connected components of a graph with $\leq b$ vertices. For setting up this graph, we have to inspect at most $\alpha * b$ variable occurrences in an s-t tgd. The cost of an application of Rule 3 is thus in $O(\alpha b^2)$.

To apply Rule 4 to an s-t tgd τ_ℓ in the current set Σ', we compare $\tau_\ell\colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y})\psi(\mathbf{x},\mathbf{y})$ with every $\tau_i \in \Sigma'$, such that $i \neq \ell$. With $\tau_i\colon \varphi_i(\mathbf{x}_i) \rightarrow (\exists \mathbf{y}_i)\psi_i(\mathbf{x}_i,\mathbf{y}_i)$, we proceed as follows:

1. For each $i \neq \ell$, compute all possible substitutions λ_{ij}, s.t. $At(\varphi_i(\mathbf{x}_i\lambda_{ij})) \subseteq At(\varphi(\mathbf{x}))$.
   Every such λ_{ij} is uniquely determined by an assignment of the $\leq b$ atoms of $\varphi_i(\mathbf{x}_i)$ to the $\leq b$ atoms of $\varphi(\mathbf{x})$. Hence, for every i, there are at most b^b possible substitutions λ_{ij}.
2. For all i, j, compute $\mathcal{A}_{ij} = At(\psi_i(\mathbf{x}_i\lambda_{ij}, \mathbf{y}_{ij}))$, where $\mathbf{y}_{ij}$ is a set of fresh variables, i.e., we apply the substitutions λ_{ij} computed in the first step to the conclusion of τ_i and rename the variables $\mathbf{y}_i$ apart.
3. Let $\mathcal{A} = \bigcup_{i\neq\ell}\bigcup_j \mathcal{A}_{ij}$. Try to find a substitution μ, s.t. $At(\psi(\mathbf{x},\mathbf{y}\mu)) \subseteq \mathcal{A}$. If such a μ exists, delete τ_ℓ.

In Step 1, we compute all solutions of a CQ with $\leq b$ atoms over a database with $\leq b$ atoms. In total, we apply this step to at most b^2n^2 pairs (τ_ℓ, τ_i), which is feasible in total time $O(b^2n^2\alpha b^b)$. As a result, we get $\mathcal{A}$ as the union of at most $b^2n^2b^b$ sets $\mathcal{A}_{ij}$, each consisting of $\leq b$ atoms. Hence, $\mathcal{A}$ contains $\leq n^2b^{b+3}$ atoms. Step 2 is then feasible in time $O(\alpha n^2 b^{b+3})$. Finally, in Step 3, we have to evaluate a Boolean CQ with $\leq b$ atoms over a database $\mathcal{A}$ consisting of $\leq n^2b^{b+3}$ atoms. This is feasible in $O(\alpha(n^2b^{b+3})^b)$. In total, the entire computation required for an application of Rule 4 thus fits into time $O(||\Sigma||^{f(b)})$ for some function $f(.)$, which depends only on b but not on the size of the input.

An application of Rule 5 is very similar to Rule 4. The first two steps above are identical. Only in Step 3 we do not search for a μ with $At(\psi(\mathbf{x},\mathbf{y}\mu)) \subseteq \mathcal{A}$. At this stage, we know that such a μ does not exist. Instead, we search for a μ, s.t. $At(\psi(\mathbf{x},\mathbf{y}\mu)) \subseteq \mathcal{A} \cup At(\psi(\mathbf{x},\mathbf{y}))$ and $At(\psi(\mathbf{x},\mathbf{y}\mu)) \cap \mathcal{A} \neq \emptyset$. If such a μ exists, we choose μ, s.t. $At(\psi(\mathbf{x},\mathbf{y}\mu)) \cap \mathcal{A}$ is maximal (w.r.t. set inclusion). Note that the desired s-t tgd τ' for replacing τ is obtained as $\tau'\colon \varphi(\mathbf{x}) \rightarrow (\exists \mathbf{y}')\psi'(\mathbf{x},\mathbf{y}')$, s.t. $At(\psi'(\mathbf{x},\mathbf{y}')) = At(\psi(\mathbf{x},\mathbf{y})) \setminus \{A \mid A \in At(\psi(\mathbf{x},\mathbf{y}))$ and $A\mu \in \mathcal{A}\}$. In other words, the set of atoms in the conclusion of τ' becomes minimal if $At(\psi(\mathbf{x},\mathbf{y}\mu)) \cap \mathcal{A}$ is maximized. Clearly, Steps 1 and 2 above do not have to be repeated for Rule 5. Step 3 of an application of Rule 5 boils down to essentially the same kind of CQ evaluation as for Rule 4. We thus end up again with an upper bound of $O(||\Sigma||^{g(b)})$ on the computation time, where $g(.)$ is a function, which depends only on b but not on the size of the input. □

D Full proof of Theorem 11

Theorem 11 *Let $\mathcal{M} = \langle \mathbf{S}, \mathbf{T}, \Sigma_{st} \cup \Sigma_t \rangle$ be a schema mapping and let $\Sigma*_s \cup \Sigma^*_{st} \cup \Sigma_t$ be the normal form of $\Sigma_{st} \cup \Sigma_t$. Moreover, let I be a source instance and J^* the canonical universal solution for I under $\mathcal{M}$ obtained via an oblivious chase with Σ^*_{st} followed by a chase with Σ_t in arbitrary order. Then J^* is unique up to isomorphism. We denote J^* as $CanSol^*(I)$.*

Proof By the equivalence of Σ, the chase with Σ fails iff the chase with Σ' fails. We may thus restrict ourselves to the case that both chases succeed. Suppose that the chase of J with Σ (resp. Σ') consists of n (resp. n') egd-applications and write J_i (resp. J'_i) to denote the intermediate result after the i-th step with $i \in \{0, \ldots, n\}$ (resp. $i \in \{0, \ldots, n'\}$). In particular, $J = J_0 = J'_0$. Clearly, every J_i and J'_i is a homomorphic image of J.

Egds have the effect that variables may disappear from J. We therefore concentrate on the *positions* in J. To this end, we assume that every atom A in J is equipped with a unique identifier $id(A)$. A position in J is thus uniquely determined by $id(A)$ of an atom A and a position in A (i.e., an index between 1 and the arity of the predicate symbol of A). We assume that duplicate atoms, which may be produced by the chase, are not deleted. Then the positions in J persist in all instances J_i and J'_i, even though variables from J may disappear in J_i and J'_i. The application of an egd $\varepsilon\colon \varphi(\mathbf{x}) \rightarrow z_1 = z_2$ to an instance J_i (resp. J'_i) means that the variables in ε are bound by some substitution $\sigma\colon \mathbf{x} \rightarrow Const \cup var(J)$. This substitution σ is determined by assigning each atom in $\varphi(\mathbf{x})$ to an atom in J_i. Thus, every variable occurrence in ε is assigned to some position in J. We can thus represent σ as a mapping from the variables in ε to tuples of positions in J: For $\mathbf{x} = \{x_1, \ldots, x_k\}$, σ is of the form $\sigma = \{x_1 \leftarrow (p_{11}, \ldots, p_{1j_1}), \ldots, x_k \leftarrow (p_{k1}, \ldots, p_{kj_k})\}$ with $j_1, \ldots, j_k \geq 1$, where the $p_{\alpha\beta}$'s denote positions in J. If a variable x_α occurs more than once in $\varphi(\mathbf{x})$ then it is mapped to several positions. Clearly, σ is well-defined for an instance J_i only if for every $\alpha \in \{1,\dots,k\}$, all positions $p_{\alpha 1},\dots,p_{\alpha j_\alpha}$ have identical values in J_i.

In order to describe the instances resulting from the application of egds to J, we introduce the notion of *equality graphs*: The vertices of these equality graphs are the positions in J. We say that an equality graph *corresponds* to an instance J' (which was obtained from J by the application of some egds) if the following equivalence holds: *Two vertices corresponding to positions p_1 and p_2 in J are connected (not necessarily adjacent) iff p_1 and p_2 have the same value in J'.* Obviously, if two instances J' and J'' obtained from J via egds are such that the corresponding equality graphs have the same connected components, then J' and J'' are isomorphic. Thus, in order to show that J^{Σ} and $J^{\Sigma'}$ are isomorphic, it suffices to show that the equality graphs corresponding to J^{Σ} and $J^{\Sigma'}$ have the same connected components.

We construct the equality graph $\mathcal{E}_i$ of J_i (and, analogously the graph $\mathcal{E}'_i$ corresponding to J'_i) inductively as follows: The vertices never change, i.e., in every graph $\mathcal{E}_i$, there is one vertex for each position in J. By slight abuse of notation, we thus identify the vertices with the positions. As edges, we introduce in the graph $\mathcal{E}_0$ corresponding to $J = J_0$ an edge between any two (vertices corresponding to) positions p_1 and p_2 if they have the same value in J. Suppose that we have already constructed J_{i-1}. Then J_i is constructed as follows: Suppose that the egd $\varphi(\mathbf{x}) \rightarrow z_1 = z_2$ with $z_1, z_2 \in \mathbf{x}$ is applied in the i-th chase step. By the above considerations, this means that we apply a substitution $\sigma = \{x_1 \leftarrow (p_{11},\dots,p_{1j_1}),\dots,x_k \leftarrow (p_{k1},\dots,p_{kj_k})\}$ to the variables $\mathbf{x} = \{x_1,\dots,x_k\}$, i.e., every variable in $\varphi(\mathbf{x})$ is mapped to one or more positions in J_{i-1} (or, equivalently, positions in J), s.t. for every $\alpha \in \{1,\dots,k\}$, all positions $p_{\alpha 1},\dots,p_{\alpha j_\alpha}$ have identical values in J_{i-1}. Note that z_j with $j \in \{1,2\}$ is a variable $x_\alpha \in \mathbf{x}$. We thus choose as vertex v_j in $\mathcal{E}_{i-1}$ some position $p_{\alpha\beta}$ for non-deterministically selected $\beta \in \{1,\dots,j_\alpha\}$. Then $\mathcal{E}_i$ is obtained from $\mathcal{E}_{i-1}$ by inserting an edge between v_1 and v_2. It can be easily verified that every $\mathcal{E}_i$ is an equality graph corresponding to J_i. By the above considerations, it suffices to show that $\mathcal{E}_n$ and $\mathcal{E}'_{n'}$ have the same connected components. We prove by induction on $i \in \{0,\dots,n'\}$ that any two vertices v_1, v_2 connected in $\mathcal{E}'_i$ are also connected in $\mathcal{E}_n$. The proof that any two vertices v_1, v_2 connected in $\mathcal{E}_i$ are also connected in $\mathcal{E}'_{n'}$ is symmetric.

"$i = 0$". $\mathcal{E}'_0$ is the initial equality graph corresponding to J. By construction, every edge in $\mathcal{E}'_0 = \mathcal{E}_0$ is contained in $\mathcal{E}_n$.

"$(i-1) \rightarrow i$". Suppose that any two vertices connected in $\mathcal{E}'_{i-1}$ are also connected in $\mathcal{E}_n$. We have to show that then also any two vertices connected in $\mathcal{E}'_i$ are connected in $\mathcal{E}_n$. By construction, $\mathcal{E}'_i$ contains at most one additional edge compared with $\mathcal{E}'_{i-1}$, say (v_1, v_2). We have to show that v_1 and v_2 are also connected (not necessarily adjacent) in $\mathcal{E}_n$. Let $\varepsilon\colon \varphi(\mathbf{x}) \rightarrow z_1 = z_2$ with $z_1, z_2 \in \mathbf{x}$ denote the egd which was applied in the i-th chase step with Σ', i.e., a substitution $\sigma = \{x_1 \leftarrow (p_{11},\dots,p_{1j_1}),\dots,x_k \leftarrow (p_{k1},\dots,p_{kj_k})\}$ was applied to the variables $\mathbf{x} = \{x_1,\dots,x_k\}$. The remainder of the proof proceeds in three steps:

(1) The substitution σ is also well-defined for J^{Σ}

(2) $At(\varphi(\mathbf{x}\sigma)) \subseteq At(J^{\Sigma})$

(3) The vertices v_1, v_2 are connected in $\mathcal{E}_n$.

Proof of (1). Let $\alpha \in \{1,\dots,k\}$. We have to show that the positions $p_{\alpha 1},\dots,p_{\alpha j_\alpha}$ have identical values in J^{Σ}. Note that σ is well-defined as a mapping of $\mathbf{x}$ to J'_{i-1} since this substitution was applied for the i-th chase step with Σ'. Hence, the vertices $p_{\alpha 1},\dots,p_{\alpha j_\alpha}$ are connected in $\mathcal{E}'_{i-1}$. But then, by the induction hypothesis, they are also connected in $\mathcal{E}_n$. This means that the positions $p_{\alpha 1},\dots,p_{\alpha j_\alpha}$ indeed have identical values in J^{Σ}.

Proof of (2). Let $A(\mathbf{x})$ be a conjunct in $\varphi(\mathbf{x})$. We have to show that $A(\mathbf{x}\sigma) \in At(J^{\Sigma})$. Clearly, $A(\mathbf{x}\sigma) \in At(J'_{i-1})$ since σ was applied for the i-th chase step with Σ', i.e., there exists an atom $B \in J$ with identifier $id(B)$, s.t. the atom B (i.e., precisely speaking, the atom with identifier $id(B)$) in J'_{i-1} coincides with $A(\mathbf{x}\sigma)$. We claim that then $A(\mathbf{x}\sigma)$ also coincides with the atom B in J^{Σ}: By definition, σ as a mapping to J^{Σ} maps the variables in $\mathbf{x}$ to the (values at the) same positions like σ as a mapping to J'_{i-1}. Hence, if a variable occurring in $A(\mathbf{x})$ is mapped to some position of B in J'_{i-1} then it is mapped to the same position of B in J^{Σ}. By (1), if some variable occurs in several positions in $A(\mathbf{x})$, then the corresponding positions of B have identical values in J^{Σ}. It thus remains to show that if some constant c occurs at a position p in A then B in J^{Σ} also has the value c at this position. Clearly, since $A(\mathbf{x}\sigma)$ coincides with B in J'_{i-1}, B has the value c at position p in J'_{i-1}, i.e., there exists a position q, s.t. p and q are connected in $\mathcal{E}'_{i-1}$ (this also comprises the case that p and q are identical) and the value at position q in J was c. Clearly, the (constant) value at position q can never change during the chase. Moreover, by the induction hypothesis, p is connected with q also in $\mathcal{E}_n$. Thus, B in J^{Σ} also has the value c at the position p.

Proof of (3). Recall the construction of $\mathcal{E}'_i$. Namely, let ε be an egd $\varphi(\mathbf{x}) \rightarrow z_1 = z_2$ with z_1 being a variable x_α and z_2 a variable x_γ. Then v_1 is some position $p_{\alpha\beta}$ and v_2 is some position $p_{\gamma\delta}$ according to the substitution σ. In other words, the edge (v_1, v_2) was introduced in $\mathcal{E}'_i$ in order to enforce the equality $z_1\sigma = z_2\sigma$ in J'_i.

We have to show that v_1, v_2 are connected in $\mathcal{E}_n$, where v_1 is the position $p_{\alpha\beta}$ and v_2 is the position $p_{\gamma\delta}$. By assumption, Σ and Σ' are equivalent. Hence, since $J^{\Sigma} \models \Sigma$, also $J^{\Sigma} \models \Sigma'$ holds. In particular, $J^{\Sigma} \models \varepsilon$. By the above considerations, $At(\varphi(\mathbf{x}\sigma)) \subseteq At(J^{\Sigma})$. Hence, the equality $z_1\sigma = z_2\sigma$ must also be fulfilled in J^{Σ} and, therefore, the values at the positions $p_{\alpha\beta}$ and $p_{\gamma\delta}$ are identical in J^{Σ}. Thus, the vertices v_1 and v_2 are indeed connected in $\mathcal{E}_n$. □