Answering Aggregate Queries in Data Exchange

Foto Afrati*
National Technical University of Athens
afrati@softlab.ece.ntua.gr

Phokion G. Kolaitis†
IBM Almaden Research Center
kolaitis@almaden.ibm.com

ABSTRACT

Data exchange, also known as data translation, has been extensively investigated in recent years. One main direction of research has focused on the semantics and the complexity of answering first-order queries in the context of data exchange between relational schemas. In this paper, we initiate a systematic investigation of the semantics and the complexity of aggregate queries in data exchange, and make a number of conceptual and technical contributions. Data exchange is a context in which incomplete information arises, hence one has to cope with a set of possible worlds, instead of a single database. Three different sets of possible worlds have been explored in the study of the certain answers of first-order queries in data exchange: the set of possible worlds of all solutions, the set of possible worlds of all universal solutions, and a set of possible worlds derived from the CWA-solutions. We examine each of these sets and point out that none of them is suitable for aggregation in data exchange, as each gives rise to rather trivial semantics. Our analysis also reveals that, to have meaningful semantics for aggregation in data exchange, a strict closed world assumption has to be adopted in selecting the set of possible worlds. For this, we introduce and study the set of the endomorphic images of the canonical universal solution as a set of possible worlds for aggregation in data exchange. Our main technical result is that for schema mappings specified by source-to-target tgds, there are polynomial-time algorithms for computing the range semantics of every scalar aggregation query, where the range semantics of an aggregate query is the greatest lower bound and the least upper bound of the values that the query takes over the set of possible worlds. Among these algorithms, the more sophisticated one is the algorithm for the average operator, which makes use of concepts originally introduced in the study of the core of the universal solutions in data exchange. We also show that if, instead of range semantics, we consider possible answer semantics, then it is an NP-complete problem to tell if a number is a possible answer of a given scalar aggregation query with the average operator.

*Part of the research on this paper was carried out while this author was visiting the IBM Almaden Research Center.
†On leave from UC Santa Cruz.

PODS'08, June 9–12, 2008, Vancouver, BC, Canada.

Categories and Subject Descriptors
H.2.5 [**Heterogeneous Databases**]: Data translation
General Terms: Algorithms, Theory
Keywords: Schema mapping, data exchange, aggregate queries

1. Introduction and Summary of Results

Data exchange, also known as data translation, can be succinctly described as the problem of transforming data structured under one schema, called the source schema, into data structured under a different schema, called the target schema, in such a way that certain constraints between the two schemas are satisfied. Data exchange is typically formalized using schema mappings between the source schema and the target schema. In recent years, the study of data exchange between relational schemas has been extensively investigated. Several different aspects of data exchange have been explored as part of this investigation. Specifically, one main direction of research has addressed the problem of identifying "good" solutions for data exchange, such as universal solutions and the core of the universal solutions, and on designing polynomial-time algorithms for producing such "good" solutions [6, 7, 13, 14]. A different main direction has explored in depth fundamental operators on schema mappings, such as the composition operator and the inverse operator [5, 8, 9, 19, 20]. A third main direction has studied the semantics and the complexity of query answering in the context of data exchange between relational schemas [6, 16, 17, 18]. Data exchange between XML schemas, as well as extensions of the framework to data exchange in the presence of arithmetic comparisons and to peer data management systems, have also been investigated [2, 4, 12].

Consider a data exchange setting specified by a schema mapping $\mathcal{M}$ between a source schema $\mathbf{S}$ and a target schema $\mathbf{T}$, and let Q be a query over $\mathbf{T}$. Query answering in this setting is the following problem: given a source instance I, find the *certain answers* of Q with respect to I. Typically, the set of *possible worlds* used in the definition of the certain answers of Q with respect to I is the set of all solutions for I. So far, the study of the certain answers in data exchange has focused primarily on conjunctive queries and on their extensions with union and inequalities $\neq$. In particular, it is known that for schema mappings specified by source-to-target tuple-generating dependencies (s-t tgds, in short), the certain answers of conjunctive queries can be computed in polynomial time in the size of the source instance I. Moreover, this tractability result extends to unions of conjunctive queries with at most one inequality per disjunct [6]. This turns out to be a sharp boundary, since computing the certain answers of conjunctive queries with at most two inequalities per disjunct is, in general, a coNP-complete problem [18]. Note that these are *data complexity* results, in the sense that both the schema mapping $\mathcal{M}$ and the query Q are fixed, i.e., only the source instance I is the input.

In this paper, we initiate an investigation of aggregate queries in data exchange. In addition to their ubiquity in on-line analytical processing (OLAP), aggregate queries are also widely used in data warehousing and in extract-transform-load (ETL) processes, two important applications that can be modeled as data exchange tasks. Given that the investigation of data exchange has been accompanied by the development of prototype systems and industrial tools [15], we believe that the time is ripe for a systematic study of aggregate query answering in data exchange. Here, we embark on this study by making a number of conceptual and technical contributions to aggregate query answering for schema mappings specified by s-t tgds.

What is the "right" semantics of aggregate queries in a data exchange setting? This is the first key issue that has to be addressed. Data exchange is a framework in which *incomplete* information arises, hence one has to cope with a set of *possible worlds*, instead of a single database. This is so because the schema mapping at hand typically under-specifies the data exchange task and, as a result, for a given source instance, there are multiple target instances (often containing null values) that satisfy the constraints of the schema mapping.

Aggregate queries have already been fruitfully studied in other contexts of incomplete information, and we can draw on that experience. Specifically, Arenas et al. [3] studied aggregate queries in the context of *inconsistent* databases, i.e., databases that violate functional dependencies or some other integrity constraints. In this case, the possible worlds are the *repairs* of the inconsistent database, which, by definition, are consistent databases that differ from the given inconsistent database in a minimal way. Arenas et al. [3] introduced *range semantics* as the semantics of aggregate queries on inconsistent database. This means that, given an inconsistent database, the aggregate query is first evaluated on each repair, and then the interval with endpoints the greatest lower bound (glb) and the least upper bound (lub) of the values obtained is returned as the answer to the aggregate query on the given inconsistent database. Range semantics have also been used in subsequent investigations of aggregate queries in inconsistent databases [10, 11]. They can be viewed as *the* certain answers for aggregate queries, since they guarantee that in every possible world, the value of the aggregate query will be in that interval. In what follows, we will adopt range semantics as the semantics of aggregate queries in data exchange.

The preceding analysis, however, is only one half of the puzzle. Before applying range semantics, one has to determine the underlying set of possible worlds. As mentioned earlier, when defining the certain answers of conjunctive queries (and, more broadly, of first-order queries) in data exchange, the set of possible worlds typically used is the set of all solutions for a given source instance. Two other alternatives, however, have also been considered. The first is the set of all universal solutions, which (unlike the set of all solutions) was shown in [7] to give rise to tractable semantics for conjunctive queries with inequalities and, more generally, for existential first-order queries. The second alternative was introduced by Libkin [17], who argued in favor of a semantics having a *closed-world-assumption* (CWA) character. To this effect, Libkin introduced the class of *CWA-solutions* and used them to define a certain answer semantics based on these solutions, as well as variants of certain answers.

We examine these three alternatives and point out that none is a viable choice as the set of possible worlds for aggregate queries in data exchange. Indeed, each of them gives rise to rather trivial range semantics for aggregate queries. Intuitively, the reason is that they allow for some form of *open world* assumption, which, in turn, renders the range semantics meaningless. Even the semantics proposed by Libkin [17] suffer from this drawback because queries are ultimately evaluated over databases that may contain arbitrary constants (these are the databases in the sets $\mathrm{Rep}(T)$, where T is a CWA-solution).

After exploring and rejecting some additional alternatives, we arrive at the conclusion that a strict closed world assumption has to be adopted for the semantics of aggregate queries in data exchange. To this effect, we propose the set of the *endomorphic images of the canonical universal solution* as the set of possible worlds for aggregate queries. This set of possible worlds gives rise to meaningful range semantics for aggregate queries. Moreover, for conjunctive queries, the certain answers with respect to this set of possible worlds coincide with the "standard" certain answers for conjunctive queries in which the set of possible worlds is the set of all solutions.

Once the semantics of aggregate queries in data exchange have been defined, we investigate the data complexity of aggregate queries of the form SELECT f FROM R, where f is one of the aggregate operators $\min(A)$, $\max(A)$, $\mathrm{count}(A)$, $\mathrm{sum}(A)$, $\mathrm{avg}(A)$, and $\mathrm{count}(*)$, and where R is a target relation symbol and A is an attribute of R. These are precisely the queries studied by Arenas et al. in [3] in the context of inconsistent databases, where they were called *scalar aggregation queries*. Our main technical result is that if $\mathcal{M}$ is schema mapping specified by s-t tgds, then there are polynomial-time algorithms for the range semantics of every scalar aggregation query of the above form. The polynomial-time algorithms for $\min$, $\max$, count, and for a special case of sum are relatively straightforward, but make use of the canonical universal solution and its core (and of known polynomial-time algorithms for computing the canonical universal solution and its core). The polynomial-time algorithm for avg is by far more sophisticated and makes use of the concepts of a *block of nulls* and of a *local endomorphism*, which were introduced in [7] and used to design a polynomial-time algorithm for computing the core of the canonical universal solution. This tractability result contrasts sharply with one of the main results in [3] to the effect that it is a coNP-hard problem to compute the range semantics of aggregate queries with the avg operator on inconsistent databases with two functional dependencies.

As mentioned earlier, range semantics are a form of certain answer semantics. Our final result asserts that the boundary between tractability and intractability for aggregate queries with the avg operator is crossed if, instead of range semantics, we consider *possible* answer semantics. Specifically, we show that it is an NP-complete problem to tell whether or not a number is a possible answer of a given scalar aggregation query with the average operator.

2. Preliminaries

In this section, we establish terminology and notation that will be used in the rest of the paper, and present a minimum amount of the necessary background material. For more details, see [3, 6, 7].

Schemas and Instances A *(relational) schema* $\mathbf{R}$ is a finite sequence $(R_1, \ldots, R_k)$ of relation symbols, each of a fixed arity. An *instance* I over $\mathbf{R}$ is a sequence $(R_1^I, \ldots, R_k^I)$, where each R_i^I is a finite relation of the same arity as R_i. We shall often use R_i to denote both the relation symbol and the relation R_i^I that interprets it. We assume that all instances are finite, which means that the relations R_i^I, $1 \leq i \leq k$, are finite. An *atom over* $\mathbf{R}$ is a formula $P(v_1, \ldots, v_n)$, where P is a relation symbol in $\mathbf{R}$ and $v_1, \ldots, v_n$ are variables. A *fact in an instance* I is an expression of the form $R_j^I(a_1, \ldots, a_n)$, where $j \leq k$ and the tuple $(a_1, \ldots, a_n)$ is a member of the relation R_j^I of I.

In what follows, we assume that $\mathbf{S}$ is a fixed *source* schema and $\mathbf{T}$ is a fixed *target* schema. We also assume that we have an infinite set Const of constants and an infinite set Null of nulls that is disjoint from Const. Since we are interested in aggregate queries, we assume that Const is a superset of the set of all (non-negative and negative) integers. Analogous results hold if a different infinite ordered set is considered. All individual values in source instances are assumed to

be constants. In contrast, target instances typically have individual values from Const ∪ Null. This situation arises when we perform data exchange from $\mathbf{S}$ to $\mathbf{T}$: the individual values of source instances are known, while incomplete information in the specification of data exchange may give rise to null values in the target instances.

Schema mappings, universal solutions, and cores A schema mapping is a triple $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ consisting of a source schema $\mathbf{S}$, a target schema $\mathbf{T}$, and a set Σ of database dependencies that specify the relationship between the source schema and the target schema. We say that $\mathcal{M}$ is *specified by* Σ.

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping. If I a is source instance, then a *solution for I under $\mathcal{M}$* is a target instance J such that $(I, J) \models \Sigma$. The set of all solutions for I under $\mathcal{M}$ is denoted by $\mathrm{Sol}(\mathcal{M}, I)$ or, simply, $\mathrm{Sol}(I)$ if $\mathcal{M}$ is understood from the context.

Let J, J' be two target instances. A function h from Const ∪ Null to Const ∪ Null is a *homomorphism* from J to J' if the following two conditions hold:

(1) For every c in Const, we have that $h(c) = c$.

(2) For every relation symbol R in $\mathbf{T}$ and every tuple $(a_1, \ldots, a_n) \in R^J$, we have that $(h(a_1), \ldots, h(a_n)) \in R^{J'}$.

Two instances J and J' are said to be *homomorphically equivalent* if there are homomorphisms from J to J' and from J' to J. An *endomorphism* of J is a homomorphism from J to J.

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping and I a ground instance. A *universal solution for I under $\mathcal{M}$* is a solution J for I under $\mathcal{M}$ such that for every solution J' for I under $\mathcal{M}$, there is a homomorphism $h : J \to J'$. Intuitively, universal solutions are the "most general" solutions among the space of all solutions for I. The set of all universal solutions for I under $\mathcal{M}$ is denoted by $\mathrm{USol}(\mathcal{M}, I)$ or, simply, $\mathrm{USol}(I)$ if $\mathcal{M}$ is understood from the context. Clearly, if J and J' are universal solutions for I, then J and J' are homomorphically equivalent.

A *source-to-target tuple-generating dependency* (or an *s-t tgd*) is a first-order formula of the form $\forall \mathbf{x}(\varphi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}))$, where $\varphi(\mathbf{x})$ is a conjunction of atoms over $\mathbf{S}$, $\psi(\mathbf{x}, \mathbf{y})$ is a conjunction of atoms over $\mathbf{T}$, and every variable in $\mathbf{x}$ occurs in an atom in $\varphi(\mathbf{x})$. Usually, we drop the universal quantifiers in the front of such an s-t tgd, and simply write $\varphi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$.

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a fixed schema mapping specified by a finite set Σ of s-t tgds. Given a source instance I, a *canonical* universal solution for I under $\mathcal{M}$ can be obtained in time polynomial in the size of I using the *naive chase* procedure. Specifically, for every s-t tgd $\varphi(\mathbf{x}) \to \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$ in Σ and for every pair of tuples $\mathbf{a}$, $\mathbf{b}$ from I such that $I \models \varphi(\mathbf{a}, \mathbf{b})$, we introduce a fresh tuple of distinct nulls $\mathbf{u}$ and create new facts in the canonical universal solution so that $\psi(\mathbf{a}, \mathbf{u})$ holds. Note that a canonical universal solution for I under $\mathcal{M}$ is unique up to renaming nulls. Thus, we refer to *the canonical universal solution for I under $\mathcal{M}$*, and denote it by $\mathrm{CanSol}(\mathcal{M}, I)$ or, simply, $\mathrm{CanSol}(I)$ if $\mathcal{M}$ is understood from the context. As an example, let $\mathcal{M}$ be the schema mapping specified by the s-t tgd

$$(x, y) \to \exists z_1 \exists z_2 (F(x, z_1) \wedge F(z_1, y) \wedge P(z_2)).$$

Consider the source instance $I = \{E(a, b), E(a, c)\}$. Then the canonical universal solution $\mathrm{CanSol}(\mathcal{M}, I)$ for I is the target instance

$$\{F(a, u_1), F(u_1, b), P(u_2), F(a, u_3), F(u_3, c), P(u_4)\},$$

where u_1, u_2, u_3, and u_4 are distinct nulls.

Let J be a target instance. A sub-instance J^* of J is called a *core* of J if there is a homomorphism h from J to J^*, but there is no homomorphism from J to a proper sub-instance J' of J^*. The following facts are well known (see [7] for the proofs):

- Every instance J has a core (this uses the finiteness of J).
- If J_1 and J_2 are cores of J, then J_1 and J_2 are isomorphic; hence, we can talk (up to isomorphism) about *the* core of J, and write $\mathrm{core}(J)$ to denote it.
- If J and J' are homomorphically equivalent target instances, then the cores of J and J' are isomorphic. In particular, if $\mathcal{M}$ is a schema mapping and I is a source instance, then all universal solutions for I have isomorphic cores. Thus, we can talk (up to isomorphism) about *the core of the universal solutions for I*.
- Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a fixed schema mapping such that Σ is a set of s-t tgds. If I is a source instance, then the core of the universal solutions for I is also a solution; hence, it is the *smallest* universal solution. Moreover, the core of the universal solutions can be computed in time polynomial in the size of I.

Continuing with the preceding example, the core of the universal solutions for the source instance $I = \{E(a, b), E(a, c)\}$ is the target instance $\{F(a, u_1), F(u_1, b), F(a, u_3), F(u_3, c), P(u_2)\}$. Another isomorphic core is $\{F(a, u_1), F(u_1, b), F(a, u_3), F(u_3, c), P(u_4)\}$.

Certain Answers and Aggregate Certain Answers Let $\mathbf{R}$ and $\mathbf{R}^*$ be two (not necessarily distinct) relational schemas. Suppose that for every $\mathbf{R}$-instance I, there is a set $\mathcal{W}(I)$ of $\mathbf{R}^*$-instances that are associated with I; intuitively, we view $\mathcal{W}(I)$ as a set of *possible worlds* associated with I. For example, in the context of data exchange, $\mathcal{W}(I)$ may the set of all solutions for I; similarly, in the context of inconsistent databases, $\mathcal{W}(I)$ may be the set of all repairs of I.

DEFINITION 2.1. *Let Q be a k-ary first-order query over $\mathbf{R}^*$.*

- *We say that a k-tuple $\mathbf{t}$ is a* certain answer of Q with respect to I and $\mathcal{W}(I)$ *if for every $J \in \mathcal{W}(I)$, we have that $\mathbf{t} \in Q(J)$, where $Q(J)$ is the k-ary relation obtained by evaluating Q on J.*
- *We write $\mathrm{certain}(Q, I, \mathcal{W}(I))$ to denote the set of all certain answers of Q with respect to I and $\mathcal{W}(I)$. In symbols,*

$$\mathrm{certain}(Q, I, \mathcal{W}(I)) = \bigcap \{Q(J) : J \in \mathcal{W}(I)\}.$$

Assume that Q is a k-ary first-order query on $\mathbf{R}^*$ and f is one of the aggregate operators $\min(A)$, $\max(A)$, $\mathrm{count}(A)$, $\mathrm{sum}(A)$, $\mathrm{avg}(A)$, and $\mathrm{count}(*)$, where A is an attributes of Q. In what follows, we write $f(Q)$ to denote the aggregate query SELECT f FROM Q.

What are the "certain answers" of the aggregate query $f(Q)$ with respect to I and the set of possible worlds $\mathcal{W}(I)$? One natural interpretation is to consider the *range* of all possible values $f(Q)(J)$, where J is in $\mathcal{W}(I)$, and then return the interval with endpoints the glb and the lub of these possible values. As mentioned earlier, this is precisely the semantics for aggregate queries in inconsistent databases adopted by Arenas et al. [3].

DEFINITION 2.2. *Let Q be a k-ary first-order query over $\mathbf{R}^*$ and let f be one of the aggregate operators $\min(A)$, $\max(A)$, $\mathrm{count}(A)$, $\mathrm{sum}(A)$, $\mathrm{avg}(A)$, and $\mathrm{count}(*)$, where A is an attributes of Q. When applying aggregate operators to instances with null values, we adopt the convention of SQL for the treatment of nulls. Specifically, for all aggregate operators other than $\mathrm{count}(*)$, a null-elimination step is performed before the aggregate operator is applied, hence null values are not taken into account into the computation (see [1]).*

- *We say that a value r is a* possible answer of Q with respect to I and $\mathcal{W}(I)$ *if there is an instance J in $\mathcal{W}(I)$ such that $f(Q)(J) = r$.*
- *We write $\mathrm{poss}(f(Q), I, \mathcal{W}(I))$ to denote the set of all possible answers of the aggregate query $f(Q)$ with respect to I and $\mathcal{W}(I)$.*
- *The* aggregate certain answers of the aggregate query $f(Q)$ with respect to I and $\mathcal{W}(I)$ *is the interval*

$$[\mathrm{glb}(\mathrm{poss}(f(Q), I, \mathcal{W}(I))), \mathrm{lub}(\mathrm{poss}(f(Q), I, \mathcal{W}(I)))],$$

where glb *and* lub *stand, respectively, for greatest lower bound and least upper bound.*

• *We write* $\text{agg-certain}(f(Q), I, \mathcal{W}(I))$ *to denote the aggregate certain answers of* $f(Q)$ *with respect to* I *and* $\mathcal{W}(I)$*. If the set* $\mathcal{W}(I)$ *of possible worlds is understood from the context, then we simply write* $\text{agg-certain}(f(Q), I)$.

Clearly, the aggregate certain answers provide the following guarantee: if $\text{agg-certain}(f(Q), I, \mathcal{W}(I)) = [c, d]$, then for every J in $\mathcal{W}(I)$, we have that $c \leq f(Q)(J) \leq d$.

We will study aggregate certain answers in the context of data exchange, where, as seen earlier, we typically have instances containing null values. Note that the null-elimination step that precedes the application of aggregate operators can have an impact on the results. For example, let $R = \{(101, 101), (201, 201, (301, 301), (u, 0)\}$ be a binary relation with attributes A and B, where u is a null. Then $\text{avg}(A) = 201$ and $\text{avg}(B) = 150.75$. In contrast, if nulls were not eliminated, then both averages would be equal to 150.75.

3. Semantics of Aggregation in Data Exchange

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping such that Σ is a finite set of s-t tgds. Consider an aggregate query of the form $f(Q)$, where Q is a k-ary first-order query over the target schema $\mathbf{T}$ and f is an aggregate operator. In view of the preceding discussion, to assign meaningful semantics to the query $f(Q)$ in this context, we must first associate a set $\mathcal{W}(I)$ of suitable possible worlds with every source instance I, and then return the interval $\text{agg-certain}(Q, I, \mathcal{W}(I))$ as the semantics of $f(Q)$ in this context.

In data exchange, three different sets of possible worlds have been considered in the study of the semantics of first-order queries: the set $\text{Sol}(\mathcal{M}, I)$ of all solutions for I [6]; the set $\text{USol}(\mathcal{M}, I)$ of all universal solutions for I [7]; and a set of possible worlds based on the CWA-solutions [17]. We now examine each of these three alternatives and show that none of them is suitable as a set of possible worlds for the semantics of aggregate queries.

3.1 $\text{Sol}(I)$ and $\text{USol}(I)$ as Sets of Possible Worlds

The set of all solutions and the set of all universal solutions give rise to rather trivial aggregate certain answer semantics. This is a consequence of the following two simple results.

PROPOSITION 3.1. *Let* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping such that* Σ *is a finite set of s-t tgds. Assume that* R *is a target relation symbol and* f *is one of the aggregate operators* $\min(A)$, $\max(A)$, $\text{sum}(A)$, *and* $\text{avg}(A)$, *where* A *is an attribute of* R*. If* I *is source instance, then* $\text{agg-certain}(f(R), I, \text{Sol}(I)) = (-\infty, \infty)$.

PROOF. *(Sketch)* We use the fact that if J is a target instance that contains the canonical universal solution $\text{CanSol}(I)$, then J is a solution for I (this is true because Σ is a set of s-t tgds). Consequently, we can obtain solutions with an arbitrary number of tuples and with arbitrary positive and negative integers as values for the attribute A. □

PROPOSITION 3.2. *Let* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping such that* Σ *is a finite set of s-t tgds. Assume that* R *is a target relation symbol with at least two attributes, and* A *is an attribute of* R*. Let* I *be a source instance, let* $a = \min(R.A)(\text{CanSol}(I))$, *and let* $b = \max(R.A)(\text{CanSol}(I))$. *Then the following are true:*

1. $\text{agg-certain}(\min(R.A), I, \text{USol}(I)) = a$.
2. $\text{agg-certain}(\max(R.A), I, \text{USol}(I)) = b$.
3. *If* $a = b$, *then* $\text{agg-certain}(\text{avg}(R.A), I, \text{USol}(I)) = a$.
4. *If* $a < b$, *then* $\text{agg-certain}(\text{avg}(R.A), I, \text{USol}(I)) = (a, b)$.
5. *If* $\text{CanSol}(I)$ *contains a fact* $R(\mathbf{t})$ *in which* $\mathbf{t}[A]$ *is a positive integer and a fact* $R(\mathbf{t}')$ *in which* $\mathbf{t}'[A]$ *is a negative integer, then* $\text{agg-certain}(\text{sum}(R.A), I, \text{USol}(I)) = (-\infty, \infty)$.

PROOF. *(Sketch)* Homomorphisms map constants to themselves, and all universal solutions are homomorphically equivalent to each other. Consequently, all universal solutions have the same set of constants in every attribute; in particular, the minimum and the maximum of an attribute in an arbitrary universal solution coincide, respectively, with a and b. For simplicity, assume that A is the first attribute of R. Assume that the source instance I is such that $\text{CanSol}(I)$ contains the fact $R(b, p_2, \ldots, p_k)$. Let m be a positive integer and let u_i^j be nulls, where $2 \leq i \leq k$ and $1 \leq j \leq m$. It is easy to see that if we add the facts $R(b, u_2^j, \ldots, u_k^j)$, $1 \leq j \leq m$, to $\text{CanSol}(I)$, then the resulting target instance can be mapped homomorphically into $\text{CanSol}(I)$, hence it is a universal solution for I. From this, it follows that $\text{lub}(\text{poss}(\text{avg}(R.A), I, \text{USol}(I))) = b$. Moreover, if $b > 0$, then $\text{lub}(\text{poss}(\text{sum}(R.A), I, \text{USol}(I))) = \infty$. The argument for the greatest lower bound is similar. □

3.2 $\text{Rep}(\text{CanSol}(I))$ as a Set of Possible Worlds

Both the set of possible worlds of all solutions and the set of possible worlds of all universal solutions have an *open-world-assumption* (OWA) character. Libkin [17] argued in favor of a semantics of first-order queries in data exchange that have a *closed-world-assumption* (CWA) character, and proceeded to propose such a semantics. Libkin's semantics are defined in two steps that we now summarize.

The first step is to introduce the concept of *CWA-solutions* in data exchange. Libkin made a case that CWA-solutions are *good* solutions for data exchange purposes because they satisfy certain requirements, which, intuitively, assert that "every fact in the target instance is directly justified by the source instance and the s-t tgds." Instead of reproducing the rather elaborate definition of a CWA-solution, we will use the following characterization, given in Theorem 3.4 of [17].

THEOREM 3.3. *Let* $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ *be a schema mapping such that* Σ *is a set of s-t tgds. The following two statements are equivalent for a source instance* I *and a target instance* J.

1. J *is a CWA-solution for* I.
2. J *is a homomorphic image of* $\text{CanSol}(I)$*; moreover, there is a homomorphism from* J *to* $\text{CanSol}(I)$.[1]

We write $\text{CWA}(I)$ to denote the set of all CWA-solutions for I. Note that both $\text{CanSol}(I)$ and $\text{core}(\text{CanSol}(I))$ are CWA-solutions for I. Before proceeding any further, we should point out that Libkin [17] considered a slightly different concept of homomorphism than the more standard one we use here. Specifically, he considered *null-preserving* homomorphisms, i.e., homomorphisms that map nulls to nulls, while the homomorphisms we use here may map nulls to nulls or to constants (constants map to themselves under both concepts). As stated in [17], all the results in that paper have "exact analogs" when the more standard concept of homomorphism is used.

The second step is to associate, with every CWA-solution J for I, a set $\text{Rep}(J)$ of null-free target instances that is obtained as follows. A *valuation* v is a mapping $v : \underline{\text{Null}} \rightarrow \underline{\text{Const}}$ from the set of nulls to the set of constants. If J is a target instance and v is a valuation, then $v(J)$ is the null-free target instance obtained from J by replacing every null u in J by the constant $v(u)$. Then $\text{Rep}(J)$ is defined as

$$\text{Rep}(J) = \{v(J) : v \text{ is a valuation}\}.$$

$\text{Rep}(J)$ coincides with the set of null-free homomorphic images of J. Using these concepts, Libkin [17] introduced the following version of certain-answer semantics for first-order queries in data exchange.

[1] Actually, Libkin stipulated a third condition, namely, that J contains $\text{core}(\text{CanSol}(I))$. It is easy to see, however, that this third condition is superfluous.

DEFINITION 3.4. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping such that Σ is a set of s-t tgds. If Q is a k-ary first-order query and I is a source instance, then*

$$\text{certain}_{\Box}(Q, I) = \bigcap_{J \in \text{CWA}(I)} \Big(\bigcap_{J' \in \text{Rep}(J)} Q(J') \Big).$$

In words, a k-tuple $\mathbf{t}$ is in $\text{certain}_{\Box}(Q, I)$ if and only if for every CWA-solution J for I, and for every target instance J' in $\text{Rep}(J)$, we have that $\mathbf{t} \in Q(J')$.

It is easy to see that $\text{certain}_{\Box}(Q, I)$ is a special case of the concept $\text{certain}(Q, I, \mathcal{W}(I))$ introduced in Definition 2.1. Indeed,

$$\text{certain}_{\Box}(Q, I) = \text{certain}(Q, I, \bigcup_{J \in \text{CWA}(I)} \text{Rep}(J)).$$

Moreover, it is also easy to verify that

$$\bigcup_{J \in \text{CWA}(I)} \text{Rep}(J) = \text{Rep}(\text{CanSol}(I))$$

and, consequently,

$$\text{certain}_{\Box}(Q, I) = \text{certain}(Q, I, \text{Rep}(\text{CanSol}(I)).$$

This last fact was already obtained by Libkin (see [17, Theorem 4.1]). Thus, $\text{certain}_{\Box}(Q, I)$ is the special case of $\text{certain}(Q, I, \mathcal{W}(I))$ in which $\mathcal{W}(I) = \text{Rep}(\text{CanSol}(I))$.

If Q is an arbitrary first-order query, then $\text{certain}(Q, I, \text{Sol}(I))$, $\text{certain}(Q, I, \text{USol}(I))$, and $\text{certain}_{\Box}(Q, I)$ may differ from each other. If Q, however, Q is a conjunctive query, then

$$\text{certain}(Q, I, \text{Sol}(I)) = \text{certain}(Q, I, \text{USol}(I)) = \text{certain}_{\Box}(Q, I).$$

In this case, $\text{certain}(Q, I, \text{Sol}(I))$ can be obtained by first evaluating the conjunctive query Q on $\text{CanSol}(I)$ and then removing all tuples containing at least one null. This implies that the data complexity of the certain answers of conjunctive queries is in PTIME (see [6, 7, 17]). Similar results hold for unions of conjunctive queries.

At this point, it is natural to consider the set $\text{Rep}(\text{CanSol}(I))$ as the next candidate for the set of possible worlds for the semantics of aggregate queries in data exchange. In other words, it is natural to consider $\text{agg-certain}(f(Q), I, \text{Rep}(\text{CanSol}(I))$ as the semantics of aggregation in data exchange, where f is an aggregate operator and Q is a first-order query. It does not take long to realize, however, that $\text{agg-certain}(f(Q), I, \text{Rep}(\text{CanSol}(I))$ suffers essentially from the same shortcomings as $\text{agg-certain}(f(Q), I, \text{Sol}(I))$.

PROPOSITION 3.5. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping such that Σ is a finite set of s-t tgds. Assume that R is a target relation symbol and f is one of the aggregate operators $\min(A)$, $\max(A)$, $\text{sum}(A)$, and $\text{avg}(A)$, where A is an attribute of R. If I is source instance such that $\text{CanSol}(I)$ contains at least one fact $R(\mathbf{t})$ in which $\mathbf{t}[A]$ is a null, then*

$$\text{agg-certain}(f(R), I, \text{Rep}(\text{CanSol}(I)) = (-\infty, \infty).$$

PROOF. Since the canonical universal solution $\text{CanSol}(I)$ for I contains at least one fact $R(\mathbf{t})$ such that $t[A]$ is a null, it follows that $\text{Rep}(\text{CanSol}(I))$ contains instances in which the attribute A has arbitrarily small and arbitrarily large values. □

In many respects, the failure of Libkin's approach to yield non-trivial semantics for aggregation in data exchange is due to the fact that, after all, this approach *deviates* from the closed world assumption to a large extent. This is caused by the sets $\text{Rep}(J)$, where J varies over all CWA-solutions for I. Indeed, while the concept of a CWA-solution adheres to the closed world assumption, the sets $\text{Rep}(J)$ do *not*, since they consist of *arbitrary* null-free homomorphic images of CWA-solutions. In particular, the target instances in $\text{Rep}(J)$ may contain constants that are justified neither by the source instance at hand, nor by the s-t tgds of the given schema mapping.

The preceding analysis suggests that, in order to obtain non-trivial semantics for aggregation in data exchange, we need to look beyond $\text{Sol}(I)$, $\text{USol}(I)$, and $\text{Rep}(\text{CanSol}(I))$ as suitable sets of possible worlds for aggregate queries in data exchange; more importantly, we need to adopt a rather strict closed world assumption.

At first sight, it seems that a way to overcome the shortcomings of Libkin's semantics is to consider, for every CWA-solution J for I, the subset $\text{Rep}^*(J)$ of $\text{Rep}(J)$ that consists of all target instances J' of the form $J' = v(J)$, where $v : \underline{\text{Null}} \rightarrow \underline{\text{Const}}$ is a valuation that maps nulls to constants occurring in the source instance I. In other words, $\text{Rep}^*(J)$ is the set of all null-free homomorphic images of J in which every value is a constant in the source instance I. This approach gives rise to the set $\bigcup_{J \in \text{CWA}(I)} \text{Rep}^*(J)$ as a candidate set of possible worlds for aggregation in data exchange. As before,

$$\bigcup_{J \in \text{CWA}(I)} \text{Rep}^*(J) = \text{Rep}^*(\text{CanSol}(I)).$$

Thus, we are led to consider $\text{Rep}^*(\text{CanSol}(I)$ as a set of possible worlds, and $\text{agg-certain}(f(Q), I, \text{Rep}^*(\text{CanSol}(I))$ as candidate semantics of aggregate queries in data exchange.

It is easy to see that $\text{agg-certain}(f(Q), I, \text{Rep}^*(\text{CanSol}(I))$ is non-trivial semantics. Nonetheless, this semantics suffers from a different serious drawback. Specifically, if $\text{Rep}^*(\text{CanSol}(I))$ is used as a set of possible worlds for the semantics of conjunctive queries Q, then $\text{certain}(Q, I, \text{Rep}^*(\text{CanSol}(I))$ may *differ* from the "standard" (and robust) semantics $\text{certain}(Q, I, \text{Sol}(I))$ (recall that, for conjunctive queries, the latter coincides with $\text{certain}(Q, I, \text{USol}(I))$ and with $\text{certain}_{\Box}(Q, I)$).

EXAMPLE 3.6. *Let $\mathcal{M}$ be the schema mapping specified by the s-t tgds $E(x, y) \rightarrow F(x, y) \wedge P(x) \wedge P(y)$ and $U(x) \rightarrow \exists y B(y)$, and let Q be the unary conjunctive query $P(x) \wedge \exists y \exists z (F(y, z)) \wedge B(z))$. Consider the source instance $I = \{E(1, 2), E(2, 1), U(1), U(2)\}$. Then $\text{CanSol}(I) = \{F(1, 2), F(2, 1), P(1), P(2), B(u_1), B(u_2)\}$, where u_1 and u_2 are distinct nulls. It is clear that*

$$\text{certain}(Q, I, \text{Sol}(I)) = Q(\text{CanSol}(I)) = \emptyset.$$

In contrast, if $J \in \text{Rep}^*(\text{CanSol}(I))$, then $Q(J) = \{1, 2\}$; hence $\text{certain}(Q, I, \text{Rep}^*(\text{CanSol}(I)) = \{1, 2\}$. □

3.3 Endomorphic Images of $\text{CanSol}(I)$ as Possible Worlds

In Sections 3.1 and 3.2, we examined four different candidate sets of possible worlds for the semantics of aggregate queries in data exchange, and determined that none of them is viable. One inevitable conclusion drawn from this analysis is that we must adopt an approach that is based on a strict closed world assumption. Towards this goal, we now consider the set of possible worlds formed by the endomorphic images of the canonical universal solution.

DEFINITION 3.7. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping such that Σ is a set of s-t tgds. If I is a source instance, then we write $\text{Endom}(I, \mathcal{M})$ for the set of all endomorphic images of the canonical universal solution $\text{CanSol}(I)$ for I. If $\mathcal{M}$ is understood from the context, then we simply write $\text{Endom}(I)$ in place of $\text{Endom}(I, \mathcal{M})$.*

The following are some basic properties of the set $\text{Endom}(I)$.

- $\text{Endom}(I)$ contains both $\text{CanSol}(I)$ and $\text{core}(\text{CanSol}(I))$ as members.
- Every member of $\text{Endom}(I)$ is a sub-instance of $\text{CanSol}(I)$; the converse, however, need not hold (see [17]).

- Every member of $\mathrm{Endom}(I)$ is a CWA-solution for I; the converse, however, need not hold.

We propose to use $\mathrm{Endom}(I)$ as the set of possible worlds for the semantics of aggregate queries in data exchange. There are three main reasons for this choice: (i) the members of $\mathrm{Endom}(I)$ adhere to a strict closed world assumption; (ii) if $\mathrm{Endom}(I)$ is used as the set of possible worlds for the semantics of conjunctive queries Q, then $\mathrm{certain}(Q, I, \mathrm{Endom}(I))$ coincides with $\mathrm{certain}(Q, I, \mathrm{Sol}(I))$; and (iii) $\text{agg-certain}(f(Q), I, \mathrm{Endom}(I))$ is non-trivial semantics for aggregate queries $f(Q)$. Next, we elaborate on each of these reasons.

The basic properties of $\mathrm{Endom}(I)$ imply that a target instance is a member of $\mathrm{Endom}(I)$ if and only if it is both a CWA-solution for I and a sub-instance of the canonical universal solution $\mathrm{CanSol}(I)$. This is the precise sense in which the members of $\mathrm{Endom}(I)$ adhere to a strict closed world assumption: since they are CWA-solutions, they satisfy the stringent conditions stipulated by Libkin [17], which assert that every fact in them has a tight justification by the source instance and the s-t tgds of the schema mapping; moreover, since they are sub-instances of $\mathrm{CanSol}(I)$, they do not contain any facts that are not already produced by the naive chase procedure.

PROPOSITION 3.8. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping in which Σ is a set of s-t tgds.*

- *If Q is a union of conjunctive queries over $\mathbf{T}$ and I is a source instance, then*
$$\mathrm{certain}(Q, I, \mathrm{Endom}(I)) = \mathrm{certain}(Q, I, \mathrm{Sol}(I)).$$
- *If Q is a union of conjunctive queries with inequalities $\neq$ over $\mathbf{T}$ and I is a source instance, then*
$$\mathrm{certain}(Q, I, \mathrm{Endom}(I)) = \mathrm{certain}(Q, I, \mathrm{USol}(I)).$$

PROOF. *(Sketch)* The first part holds because $\mathrm{CanSol}(I)$ is a member of $\mathrm{Endom}(I)$ and unions of conjunctive queries are preserved under homomorphisms. The second part follows from the following three facts: the core of $\mathrm{CanSol}(I)$ is a member of $\mathrm{Endom}(I)$; for every member J of $\mathrm{Endom}(I)$, there is a 1-1 homomorphism from $\mathrm{core}(\mathrm{CanSol}(I))$ to J; unions of conjunctive queries with inequalities $\neq$ are preserved under 1-1 homomorphisms. □

Finally, we give an example that demonstrates that, even for schema mappings specified by a single s-t tgd, $\mathrm{Endom}(I)$ gives rise to interesting semantics for aggregate queries in data exchange.

EXAMPLE 3.9. *Consider the schema mapping $\mathcal{M}$ specified by the s-t tgd: $P(x, y) \rightarrow \exists z(T(x, y) \wedge T(x, z))$. Let I_n, $n \geq 1$, be the source instance $\{P(a_1, b_1), \ldots, P(a_n, b_n)\}$, where the a_i's and the b_i's are positive integers. Then $\mathrm{CanSol}(I_n)$ is the target instance*

$$\{T(a_1, b_1), \ldots, T(a_n, b_n), T(a_1, u_1), \ldots, T(a_n, u_n)\},$$

where the u_i's are distinct nulls. Every subset K of $\{1, \ldots, n, \}$ gives rise to an endomorphism h_K of J_n defined as follows: $h_K(u_i) = u_i$ if i is in K, and $h_K(u_i) = b_i$ if i is not in K. Vice versa, it is easy to see that every endomorphism of J_n is determined by a subset K of $\{1, \ldots, n\}$. Thus, $\mathrm{Endom}(I)$ consists of the exponentially many endomorphic images $J_K = h_K(\mathrm{CanSol}(I_n))$, as K ranges over all subsets of $\{1, \ldots, n\}$. Note that $\mathrm{CanSol}(I_n) = J_K$ with $K = \{1, \ldots, n\}$, while $\mathrm{core}(\mathrm{CanSol}(I_n)) = J_K$ with $K = \emptyset$.

Assume that the attributes of T are A and B. It is easy to see that if $K \subseteq \{1, \ldots, n\}$, then we have that $\mathrm{count}((T.A)^{J_K}) = n + |K|$ and $\mathrm{sum}((T.A)^{J_K} = (\sum_{i=1}^{n} a_i) + (\sum_{i \in K} a_i)$. Consequently,

$$\text{agg-certain}(\mathrm{count}(T.A), I_n, \mathrm{Endom}(I_n)) = [n, 2n]$$

and

$$\text{agg-certain}(\mathrm{sum}(T.A), I_n, \mathrm{Endom}(I_n)) = [\textstyle\sum_{i=1}^{n} a_i, 2\sum_{i=1}^{n} a_i].$$

Moreover, the endpoints of these intervals are obtained by evaluating $\mathrm{count}(T.A)$ and $\mathrm{sum}(T.A)$ on $\mathrm{core}(\mathrm{CanSol}(I_n))$ and on $\mathrm{CanSol}(I_n)$. Computing the range semantics of the average, however, is more complicated; in particular, the endpoints of the interval cannot always be obtained by evaluating the average on the canonical universal solution and its core. To see this, take the source instance $I = \{(1, b_1), (2, b_2), (3, b_3)\}$. It is easy to check that $\text{agg-certain}(\mathrm{avg}(T.A), I, \mathrm{Endom}(I)) = [7/4, 9/4]$, while $\mathrm{avg}(T.A)(\mathrm{core}(\mathrm{CanSol}(I))) = 2 = \mathrm{avg}(T.A)(\mathrm{CanSol}(I))$. □

4. Queries with $\min$, $\max$, sum, count, $\mathrm{count}(*)$

The main result of this section is as follows.

THEOREM 4.1. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping in which Σ is a finite set of s-t tgds, let Q be a conjunctive query over $\mathbf{T}$, and let f be one of the aggregate operators $\min(A)$, $\max(A)$, $\mathrm{sum}(A)$, $\mathrm{count}(A)$ and $\mathrm{count}(*)$, where A is an attribute of Q.*

Then the following problem is in PTIME: given a source instance I, compute $\text{agg-certain}(f(Q), I, \mathrm{Endom}(I))$. In particular, the data complexity of the aggregate certain answers of every scalar aggregation query with $\min$, $\max$, sum, count and count^* is in PTIME.

The proof of Theorem 4.1 will make use of the next two propositions. In both propositions, we assume that $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a fixed schema mapping in which Σ is a finite set of s-t tgds, and Q is a conjunctive query over $\mathbf{T}$ such that A is one of its attributes. The first proposition asserts that if I is a source instance, then $\min$ and $\max$ queries take the same value on every instance in $\mathrm{Endom}(I)$.

PROPOSITION 4.2. *Let f be the $\min$ or the $\max$ aggregate operator, let I be a source instance, and let $J = \mathrm{CanSol}(I)$ be the canonical universal solution for I under $\mathcal{M}$. Then, for every instance $T \in \mathrm{Endom}(I)$, we have that $f(Q)(T) = f(Q)(J)$. Hence*

$$\text{agg-certain}(f(Q), I, \mathrm{Endom}(I)) \quad = \quad [f(Q)(J), f(Q)(J)].$$

PROOF. First, we consider the case in which the column of $Q(J)$ under attribute A consists entirely of nulls. In this case, for every $T \in \mathrm{Endom}(I)$, the column of $Q(T)$ under attribute A consists entirely of nulls because T is a sub-instance of J. Hence the answer is "undefined" for both $\min$ and $\max$.

Assume that the column of $Q(J)$ with attribute A contains at least one constant. We argue as follows for $\max$ ($\min$ is treated in a similar way). Let $T \in \mathrm{Endom}(I)$ be such that the column of $Q(T)$ with attribute A contains at least one attribute. Since $T \subseteq J$, the monotonicity of conjunctive queries implies that $Q(T) \subseteq Q(J)$. Hence, $\max(Q.A)(T) \leq \max(Q.A)(J)$. Let $(u_1, \ldots, u_k) \in Q(J)$ be such that $u_i = (\max Q.A)(J)$. Since there is an endomorphism $h : J \rightarrow T$, we have that $(h(u_1), \ldots, h(u_k)) \in Q(T)$. But u_i is a constant, so $u_i \leq \max(Q.A)(T)$. Hence $\max(Q.A)(J) \leq \max(Q.A)(T)$. □

The following example shows that the situation for sum, count, and $\mathrm{count}(*)$ is different.

EXAMPLE 4.3. *Let $\mathcal{M}$ be a schema mapping consisting of the following three s-t tgds:*
$P_1(x, y) \rightarrow R(x, y), P_2(x, y) \rightarrow \exists z R(x, z), P_3(x, y) \rightarrow \exists R(x, z)$.
If $I = \{P_1(5, 4), P_2(5, 1), P_3(5, 1)\}$, then the canonical universal solution is $J = \{R(5, 4), R(5, w_1), R(5, w_2)\}$; moreover, we have that $core(J) = \{R(5, 4)\}$. It is easy to see that

$$\begin{aligned} \text{agg-certain}(\mathrm{sum}(R.A), I, \mathrm{Endom}(I)) &= [5, 15] \\ \text{agg-certain}(\mathrm{count}(R.A), I, \mathrm{Endom}(I)) &= [1, 3], \end{aligned}$$

where A is the first attribute of R. □

We now show that for count, $\mathrm{count}(*)$ queries, and for a special case of sum queries, the aggregate certain answers can be obtained via evaluation on the canonical universal solution and on its core.

PROPOSITION 4.4. *Assume that I is a source instance and J is the canonical universal solution* $\mathrm{CanSol}(I)$ *for I under $\mathcal{M}$.*

1. *If f is one of the aggregate operators* count *and* $\mathrm{count}(*)$*, then*
$$\text{agg-certain}(f(Q), I, \mathrm{Endom}(I)) = [f(Q)(\mathrm{core}(J)), f(Q)(J)].$$
2. *If all numeric constants in I are non-negative integers, then*
$$\text{agg-certain}(\mathrm{sum}(Q), I, \mathrm{Endom}(I)) = [\mathrm{sum}(Q)(\mathrm{core}(J)), \mathrm{sum}(Q)(J)] \quad .$$

PROOF. *(Sketch)* The proof uses the following facts: (a) If T_1 and T_2 are cores of J, then $f(Q)(T_1) = f(Q)(T_2)$; (b) For each $T \in \mathrm{Endom}(I)$, we have that $\mathrm{core}(T) \subseteq T \subseteq J$; (c) Conjunctive queries are preserved under homomorphisms; and (d) The sum operator is monotone on sets of non-negative integers, i.e., if C and D are two sets of non-negative integers such that $C \subseteq D$, then $\sum_{i \in C} i \leq \sum_{i \in D} i$. □

For every fixed schema mapping $\mathcal{M}$ specified by s-t tgds, both the canonical universal solution for a given source instance I and its core can be computed in PTIME [6, 7]. By combining this result with Propositions 4.2 and 4.4, we derive Theorem 4.1 for the operators $\min, \max, \mathrm{count}, \mathrm{count}(*)$, and for the special case of sum in which all numeric constants in the source instance are non-negative integers. The general case for sum will be obtained using results in Section 5; the details will appear in the full paper.

5. Queries with the average operator

Assume that $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a fixed schema mapping in which Σ is a finite set of s-t tgds, R is a relation symbol in the target schema $\mathbf{T}$, and A is one of the attributes of R. In this section, we give a polynomial-time algorithm for the following problem: given a source instance I, find the aggregate certain answers

$$\text{agg-certain}(\mathrm{avg}(R.A), I, \mathrm{Endom}(I))$$

of the scalar aggregation query SELECT $\mathrm{avg}(R.A)$ FROM R.

We describe here a polynomial-time algorithm such that, given a source instance I, the algorithm finds a target instance $T \in \mathrm{Endom}(I)$ such that $\mathrm{avg}(R.A)(T) = \mathrm{glb}(\mathrm{poss}(R.A, I, \mathrm{Endom}(I)))$. In words, our algorithm finds an endomorphic image of the canonical universal solution $\mathrm{CanSol}(I)$ for I such that the average on attribute A is the minimum average on attribute A over all endomorphic images of $\mathrm{CanSol}(I)$. We present the algorithm for the case in which the values to be aggregated are positive integers. It can be extended to apply to arbitrary values by adding a large enough number to each value in the source instance so that all values under attribute A in $\mathrm{CanSol}(I)$ are positive integers; then, an optimum endomorphic image for the original instance is also optimum for the all-positive instance, and vice versa. The algorithm for finding an endomorphic image of the canonical universal solution with the maximum average is symmetrical.

As seen in Example 3.9, the set $\mathrm{Endom}(I)$ of possible worlds can be exponentially big, even if Σ consists of just a single s-t tgd. Moreover and unlike the state of affairs for sum and count seen in Proposition 4.4, the aggregate certain answers for avg *cannot* be obtained by simply evaluating the query on $\mathrm{CanSol}(I)$ and on $\mathrm{core}(\mathrm{CanSol}(I))$. Thus, a more sophisticated algorithm is needed in order to find the value of the minimum average efficiently and without an exhaustive search over $\mathrm{CanSol}(I)$. Our polynomial-time algorithm uses the concepts of a *block of nulls* and of a *local endomorphism*, which were originally introduced in [7] to design a polynomial-time algorithm for computing the core of the universal solutions for schema mappings specified by s-t tgds. Several new concepts, however, have also to be introduced before describing the main ideas in our algorithm.

5.1 Locally avg-optimal endomorphic images

DEFINITION 5.1. *Let K be a target instance.*

- *Two elements in the domain of K are* adjacent *if one of the relations of K contains a tuple in which both elements occur.*
- *The* Gaifman graph of the nulls of K *is the undirected graph such that the nodes are all the nulls of K, and there exists an edge between two nulls whenever the nulls are adjacent in K.*
- *A* block of nulls of K *(or, simply, a* block of K*) is the set of nulls in a connected component of the Gaifman graph of the nulls.*
- *If B is a block of nulls of K, then $K[B]$ denotes the sub-instance of K induced by the nulls of B and the constants of K.*
- *A* block homomorphism for a block B *is a homomorphism from $K[B]$ to K.*

PROPOSITION 5.2. *Let K be a target instance with b many blocks. Then the following statements are true.*

1. *For every block B of K and for every block homomorphism h for B, there exists a block B' of K such that the image $h(B)$ of B is contained in $B' \cup$ Const.*
2. *For each block B_i of K, let h_{B_i} be a block homomorphism for B_i, where $1 \leq i \leq b$. Then the union $\bigcup_{i=1}^{b} h_{B_i}$ of these block homomorphisms is an endomorphism of K.*
3. *Let h be an endomorphism of K. Then there is a set of block homomorphisms such that h is equal to their union.*

PROOF. The first statement is immediate from the definitions. For the second statement, note that the union $h = \bigcup_{i=1}^{b} h_{B_i}$ is a well-defined function because the blocks partition the nulls, and every homomorphism maps each constant to itself. To show that h is an endomorphism of K, take a tuple t in some relation P^K of K. All nulls occurring in $\mathbf{t}$ (if any) must belong to the same block, say B_j, of K. Since h_{B_j} is a homomorphism, we have that $h(\mathbf{t}) = h_{B_j}(\mathbf{t}) \in P^K$, as desired. For the third statement, take, for each block B of K, the restriction of h to B, which is a block homomorphism for B. The union of these block homomorphisms is equal to h. □

Let K be an instance and B be a block of K. Take a block homomorphism h_B for block B. Let h be the following endomorphism of K: h agrees with h_B on $K[B]$, and h maps every null not in B to itself. Then h is a K*-local endomorphism*, as in [7], which means that h is the identity outside B. We say that h is the K*-local endomorphism defined by* h_B.

Since $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a schema mapping specified by a finite set of s-t tgds, the following basic, but very useful, facts hold [7]:

- There is a polynomial $p(n)$ such that, for every source instance I, the number of blocks of $\mathrm{CanSol}(I)$ is bounded by $p(|I|)$, where $|I|$ is the number of facts of I.
- Let c be the maximum number of existential quantifiers $\exists \mathbf{y}$ appearing in a s-t tgd $\forall \mathbf{x}(\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y} \varphi(\mathbf{x}, \mathbf{y}))$ in Σ. If I is a source instance, then every block B of $\mathrm{CanSol}(I)$ has size at most c.

The next proposition follows easily from the second fact.

PROPOSITION 5.3. *Assume that I is a source instance, B is a block of $K = \mathrm{CanSol}(\mathcal{M}, I)$, and h_B is a block homomorphism for B. Let K' be the image of the K-local endomorphism defined by h_B. Then the number of tuples in $K - K'$ is bounded by a number that depends only on $\mathcal{M}$.*

DEFINITION 5.4. *Assume that I is a source instance and that $J = h(\mathrm{CanSol}(I))$ is an endomorphic image of* $\mathrm{CanSol}(I)$.

1. *Let B be a block of* $\mathrm{CanSol}(I)$ *and let h_B be the block homomorphism for B obtained by restricting h to* $\mathrm{CanSol}(I)[B]$. *Suppose we construct an endomorphic image J' of* $\mathrm{CanSol}(I)$ *by considering the set of block homomorphisms that constructed J, replacing h_B by some other block homomorphism h'_B for B, and keeping all other block homomorphisms. Then we say that J'* is constructed from J by a B-local change using h'_B, *or simply by a* B-local change, *and write $C(h_B \rightarrow h'_B)$ to denote this.*
2. *We say that J is* locally avg-optimal *if, for every block B of J, the following property holds: if J' is a an endomorphic image of* $\mathrm{CanSol}(I)$ *obtained from J via some B-local change, then* $\mathrm{avg}(R.A)(J') \geq \mathrm{avg}(R.A)(J)$.

We now have all the conceptual framework needed to state the key lemma that will lead to a polynomial-time algorithm for the average.

LEMMA 5.5. *Assume that I is a source instance. An endomorphic image J of $CanSol(I)$ is locally avg-optimal if and only if it is an optimal endomorphic image for the average, i.e.,*

$$\mathrm{avg}(R.A)(J) = \mathrm{glb}(\mathrm{poss}(\mathrm{avg}(R.A), I, \mathrm{Endom}(I))).$$

Before we prove this lemma, we need an additional definition and an auxiliary result.

DEFINITION 5.6. *Assume that I is a source instance, B is a block of $K = \mathrm{CanSol}(\mathcal{M}, I)$, and h_B is a block homomorphism for B. We associate with h_B two numbers, n^{h_B} and s^{h_B}, defined as follows. Let K' be the image of the K-local endomorphism defined by h_B.*

- *n^{h_B} is the number of tuples in $R^K - R^{K'}$ that do not have a null in attribute A.*
- *s^{h_B} is the sum of the values over the attribute A in $R^K - R^{K'}$.*

The pair (n^{h_B}, s^{h_B}) is called the characteristic pair of h_B.

LEMMA 5.7. *Let I be a source instance, let $J = h(\mathrm{CanSol}(I))$ be an endomorphic image of* $\mathrm{CanSol}(I)$, *and let $a = \mathrm{avg}(R.A)(J)$. For every block B of* $\mathrm{CanSol}(I)$, *let h_B be the block homomorphism obtained by restricting h to* $\mathrm{CanSol}(I)[B]$. *Then the following statements are equivalent.*

1. *J is locally avg-optimal.*
2. *For every local change $C(h_B \rightarrow h'_B)$, we have that $s^{h_B} - s^{h'_B} \geq a(n^{h_B} - n^{h'_B})$.*

PROOF. Suppose that J is locally avg-optimal. Let $\mathcal{S}$ be the set of tuples in R^J of relation R that do not have a null in attribute A, let N be the number of tuples in $\mathcal{S}$, and let S be the sum over the attribute A of the tuples in $\mathcal{S}$. Then obviously $a = S/N$. Suppose a' is the average over the attribute A in J'. Then $a' = (S + s^{h_B} - s^{h'_B})/(N + n^{h_B} - n^{h'_B})$. Since J is locally avg-optimal, we have that $a' \geq a$. Using the inequality $a' \geq a$, we can derive the inequality in the statement of the lemma. The converse is proved analogously. □

PROOF. (of Lemma 5.5) Clearly, if J is optimal for the average, then it locally avg-optimal. For the other direction, suppose that J' is another endomorphic image with a smaller average. Then J' can be produced from J by a set S of local changes (at most one local change for each block B_i). For each local change, we have two characteristic pairs (n_i, s_i) and (n'_i, s'_i), the former is the one associated with the block homomorphism used in the construction of J, and the latter is the one associated with the block homomorphism used in the construction of J'. Let S be the sum over the attribute A in the relation R^J, and let N be the number of tuples without nulls in attribute A in the relation R^J. Since J' has a smaller average, we have:

$$(S + \sum_1^b s_i - \sum_1^b s'_i)/(N + \sum_1^b n_i - \sum_1^b n'_i) < S/N,$$

hence $\sum_1^b s_i - \sum_1^b s'_i > S/N(\sum_1^b n_i - \sum_1^b n'_i)$. Since there is no local change that improves the average, we have that $s_i - s'_i \leq S/N(n_i - n'_i)$, for each i. If we sum over all i's, we get $\sum_1^b s_i - \sum_1^b s'_i \leq S/N(\sum_1^b n_i - \sum_1^b n'_i)$, which is a contradiction. □

5.2 The Algorithm

By Lemma 5.5, the search for an endomorphic image of $\mathrm{CanSol}(I)$ with minimum average reduces to the search of a locally avg-optimal endomorphic image of $\mathrm{CanSol}(I)$. Here, we give a polynomial-time algorithm for the latter task. This algorithm exploits the fact that there is a polynomial number of blocks, and each block has size bounded by a number that depends only on the fixed schema mapping $\mathcal{M}$. Before describing the algorithm and establishing its properties, however, we need to develop additional sophisticated machinery.

DEFINITION 5.8. *Let I be a source instance and let $J = \mathrm{CanSol}(I)$ be the canonical universal solution for I.*

- *For each block B of J and for each subset S_j of $R^{J[B]}$, we define a pair of numbers (s_j, n_j) as follows: s_j is the sum over the attribute A of tuples in S_j, and n_j is the number of tuples in S_j that do not have a null in attribute A.*
- *A* critical number *is an integer of the form $|(s_i - s_j)/(n_i - n_j)|$, where (s_i, n_i) and (s_j, n_j) are pairs arising from different subsets of $R^{J[B]}$, for some block B.*
- *We arrange all critical numbers in increasing order and take the intervals defined by consecutive critical numbers (including the interval defined by the smallest critical number and $-\infty$, and the one defined by the greatest critical number and $+\infty$). We call them* critical intervals.

Lemma 5.7 motivates a subroutine, which we will call *subroutine compare* and denote by compare(h_i, h_j, o, C). This subroutine compares two block homomorphisms h_i and h_j for the same block (or pair of blocks in one case) and for a critical interval C; in the output o, it puts either the "worse" of the two block homomorphisms (which will then be discarded) or "equality", if they are equivalent homomorphisms; see the description of subroutine compare for the precise statements of when two homomorphisms are equivalent or when one is discarded. We distinguish two cases, depending on whether C is an open interval or a closed interval with identical endpoints.

SUBROUTINE compare $(h_i, h_j, o, C = (a, b))$ with $a \neq b$
Note: One of a and b could be $-\infty$ or $+\infty$

Case 1. $s_i - s_j > 0$ and $n_i - n_j > 0$.
If $(s_i - s_j)/(n_i - n_j) \geq b$, then $o = h_j$. Otherwise, $o = h_i$.

Case 2. $s_i - s_j < 0$ and $n_i - n_j < 0$.
If $(s_j - s_i)/(n_j - n_i) \leq a$, then $o = h_j$. Otherwise, $o = h_i$.

Case 3. $s_i - s_j = 0$ and $n_i - n_j = 0$. Then $o = equiv$

Case 4. $s_i - s_j = 0$ and $n_i - n_j \neq 0$.
If $n_j \geq n_i$, then $o = h_i$. Otherwise, $o = h_j$.

Case 5. $s_i - s_j \neq 0$ and $n_i - n_j = 0$.
If $s_j \leq s_i$, then $o = h_i$. Otherwise, $o = h_j$.

Case 6. $s_i - s_j > 0$ and $n_i - n_j < 0$. Then $o = h_j$.

Case 7. $s_i - s_j < 0$ and $n_i - n_j > 0$. Then $o = h_i$.

SUBROUTINE compare $(h_i, h_j, o, C = [a, a])$.
Only the first two cases are different as follows:

Case 1. $s_i - s_j > 0$ and $n_i - n_j > 0$.
If $(s_i - s_j)/(n_i - n_j) > a$, then $o = h_j$. If $(s_i - s_j)/(n_i - n_j) = a$, then $o = equiv$.

Case 2. $s_i - s_j < 0$ and $n_i - n_j < 0$.
If $(s_j - s_i)/(n_j - n_i) < a$, then $o = h_i$. If $(s_i - s_j)/(n_i - n_j) = a$, then $o = equiv$.

Two block homomorphisms that come out of subroutine compare as equivalent are called *avg-equivalent*. The next lemma shows that if two homomorphisms come out of Compare $(h_i, h_j, o, C = [a, a])$ as equivalent, then they indeed compare the same way with any other homomorphism. Thus, there is no inconsistency.

LEMMA 5.9. *Assume that compare $(h_i, h_j, o, C = [a, a])$ outputs $o = equiv$. If h_k is such that compare $(h_i, h_k, o, I = [a, a])$ outputs $o = h_i$ ($o = h_k$, respectively), then compare $(h_j, h_k, o, I = [a, a])$ outputs $o = h_j$ ($o = h_k$, respectively).*

We are now ready to present our polynomial-time algorithm for finding a locally avg-optimal endomorphic image of $\mathrm{CanSol}(I)$. For every block B and every critical interval C, the algorithm constructs a *decision graph* (V, E_1, E_2) using the subroutine compare. The algorithm then decides whether or not the block B is a *dismissed block for* C or a *non-dismissed block for* C. Although the criterion for this decision is part of the algorithm, we state it separately to make the presentation of the algorithm less cumbersome.

In the decision graph (V, E_1, E_2), the set V of its nodes consists of all block homomorphisms for block B; furthermore, E_1 is a set of undirected edges and E_2 is a set of directed edges. To decide whether or not B is a dismissed block for C, we first find all (maximal) connected components of (V, E_1) and then, for each such connected component, we merge in (V, E_1) all nodes in one node n_{new}, and draw an edge to/from n_{new}, whenever there was an edge to/from a node in the connected component. Let (V_1, E_3) be the resulting graph. If (V_1, E_3) has a unique *minimal* node n_{min} (i.e., a unique sink), then B is *non-dismissed for* C; otherwise, B is *dismissed for* C. If B is non-dismissed, then we select one of the block homomorphisms for block B in the connected component represented by n_{min} and call it the *optimum homomorphism h_{opt} for block B and C*.

If C is a critical interval for which no blocks is dismissed, then the optimum homomorphisms selected for each block B are *assembled* together to produce an endomorphism g of $\mathrm{CanSol}(I)$, which is a candidate for being the locally avg-optimum endomorphism returned by the algorithm. In turn, this requires another subroutine, called *subroutine assemble*, which we will give separately.

ALGORITHM AVG-OPTIMAL
Input: Source instance I
Output: A locally avg-optimal endomorphic image of $\mathrm{CanSol}(I)$

1. For each critical interval C do:
   - For each block B, do:
     - Initialize the decision graph of B with nodes all block homomorphisms for B and no edges.
     - For each pair (h_i, h_j) of block homomorphisms for B do:
       compare(h_i, h_j, o, C).
       If $o = h_i$, then draw a directed edge in the decision graph of B from h_i to h_j.
       If $o = equiv$, then draw an undirected edge in the decision graph of B from h_i to h_j.
     - Using the decision graph for block B and interval C, decide whether block B is dismissed or non-dismissed for C; if B is non-dismissed, select an optimum homomorphism h_{opt} for B and C.
   - If at least one block is dismissed for C, then dismiss C. Otherwise,
     1. assemble $(h_1, h_2, ..., h_j, g, C)$, where $h_1, h_2, ..., h_j$ are the optimum homomorphisms for interval C.
     2. Compute the average avg of $g(CanSol(I))$ over attribute A.
     3. If avg lies in C, then keep (C, avg); otherwise dismiss interval C.
2. Consider all non-dismissed intervals and find the endomorphic image among them with the minimum average. Return this as the output of the algorithm AVG-OPTIMAL on input I.

To see how we assemble block homomorphisms, we first establish some terminology and then present the subroutine assemble.

If B is a block of $CanSol(I)$, then every block homomorphism h for B is one of two kinds: (a) an *interior block homomorphisms*, which means that $h(B) \subseteq B \cup Const$; or (b) an *exterior block homomorphisms*, which means that there is a block $B' \neq B$ such that $h(B) \subseteq B' \cup Const$ and $h(B) \cap B' \neq \emptyset$. It follows that if $h(B)$ contains only constants, then h is an interior block homomorphism. Note that all exterior block homomorphisms of a specific block B have the same characteristic pair.

We construct the *inter-block graph* GB of a target instance K as follows. The nodes of GB are the blocks of K. There is a directed edge from block B to block B' if there is an exterior block homomorphism of B that maps $K[B]$ to $K[B']$. It is easy to see that all strongly connected components of GB contain blocks B_i such that the associated sub-instances $K[B_i]$ are homomorphically equivalent. Suppose that the input to subroutine assemble is a set $\mathcal{H}$ of block homomorphisms (one for each block) with the following property: let $\mathcal{B}$ be the set of all blocks for which the associated block homomorphism in $\mathcal{H}$ is an exterior one; then $\mathcal{B}$ does not contain any strongly connected component of GB that is a sink. In this case, subroutine assemble simply builds the endomorphism in the output by taking the union of all block homomorphisms in $\mathcal{H}$. Otherwise, we consider the *problematic components*, defined as follows: a sink strongly connected component is *problematic* if it is contained in $\mathcal{B}$. Intuitively, this means that if we take the union of the homomorphisms in $\mathcal{H}$, then the endomorphic image constructed will have as a sub-instance all blocks of each problematic component.

SUBROUTINE assemble $(h_1, h_2, ..., h_j, g, C)$

- Construct the inter-block graph GB.
- For each problematic component P in GB, do:
  - For each block B in P do:
    - For each pair of interior block homomorphisms h_i, h_j of B do:
      Compare (h_i, h_j, C, o).
      If $o = h_i$ or $o = equiv$, then mark h_j.
    - Choose the winning interior homomorphism h^*_B for block B arbitrarily among the unmarked ones.
  - For each pair (B_i, B_j) of P do:
    Let B be the union of $h^*_{B_i}(J[B_i])$ and $h^*_{B_j}(J[B_j])$.
    Let h_i be a homomorphism mapping all elements of $h^*_{B_i}(J[B_i])$ to $h^*_{B_j}(J[B_j])$ and all elements of $h^*_{B_j}(J[B_j])$ to themselves.
    Let h_j be a homomorphism mapping all elements of $h^*_{B_j}(J[B_j])$ to $h^*_{B_i}(J[B_i])$ and all elements of $h^*_{B_i}(J[B_i])$ to themselves.
    $Compare(h_i, h_j, C, o)$ ***caveat: we view B as one block***.
    If $o = h_i$ or $o = equiv$, then mark B_j.
  - Choose as representative of component P one of the unmarked blocks arbitrarily.
- Change $\{h_1, h_2, ..., h_j\}$ to $\{h'_1, h'_2, ..., h'_j\}$ by replacing the block homomorphisms of the component representatives by their winning interior homomorphisms.
- Construct the endomorphism g by taking the union of $\{h'_1, h'_2, ..., h'_j\}$; return g as the output of the subroutine.

EXAMPLE 5.10. *Let us revisit the schema mapping specified by the s-t tgd $P(x, y) \rightarrow \exists z(T(x, y) \wedge T(x, z))$ in Example 3.9. For every source instance I, each block of* $\mathrm{CanSol}(I)$ *is of size one. Thus, the critical numbers are precisely the values of the attribute A. It is easy to see that, for each critical interval, the algorithm, proceeds by discarding all these values (except one, for the endomorphism to go*

through) that are to the right of the interval, and keeping all values that are to the left of the interval. Furthermore, in this case, there are no problematic components, hence subroutine assemble simply returns the union of the input block homomorphisms.

It is worth pointing out that, in this example, the problem of finding an endomorphic image with the minimum average is literally equivalent to the following combinatorial problem: given a bag S_0 of positive integers, find a sub-bag S of S_0 such that: (a) S and S_0 have the same set of distinct numbers; and (b) the average of the members of S is minimized. This shows that the problem of computing $\text{agg-certain}(\text{avg}(A), I, \text{Endom}(I))$ *is algorithmically interesting, even for seemingly very simple schema mappings $\mathcal{M}$.* □

The correctness of algorithm AVG-OPTIMAL follows from the next two propositions. The first is proved along the lines of Lemma 5.7; the proof of the second will be given in the full paper.

PROPOSITION 5.11. *Suppose that, on input $h_i, h_j, C = (a, b)$, subroutine compare returns $o = h_j$ as output. Then the following statement is true. Suppose J is an endomorphic image of* $\text{CanSol}(I)$ *such that it uses block homomorphism h_i for block B,* $\text{avg}(R.A)(J) = a$, *and $a \in I$. Suppose also that J_1 is constructed from J by the local change $C(h_i \to h_j)$. If* $\text{avg}(R.A)(J_1) = a_1$, *then $a \leq a_1$.*

Two endomorphic images of $\text{CanSol}(I)$ are *avg-equivalent* if the average over the attribute A is the same on both.

PROPOSITION 5.12. *The following statements are true for algorithm* AVG-OPTIMAL.

1. *For each critical interval C and for each block B, there is at most one block homomorphism (up to avg-equivalence) which is a sink.*
2. *There is at least one non-dismissed critical interval. Consequently, the algorithm always has a non-empty output.*
3. *Let J be the endomorphic image returned by the algorithm. Then J is a locally avg-optimal endomorphic image.*

The running time of Algorithm AVG-OPTIMAL is bounded by a polynomial in the size of the input source instance I. This uses the fact that $\text{CanSol}(I)$ has polynomially-many blocks, and each block has size bounded by a constant, which, in turn, implies that there are polynomially-many critical intervals. Combined with Proposition 5.12, this completes the proof of the main result of this section.

THEOREM 5.13. *Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping in which Σ is a finite set of s-t tgds, let R be a target relation symbol, and let A be an attribute of R. Then there is a polynomial-time algorithm for the following problem: given a source instance I, compute* $\text{agg-certain}(\text{avg}(R.A), I, \text{Endom}(I))$.

In contrast to the aggregate certain answers, computing the possible answers of scalar aggregation queries with the average operator turns out to be an NP-complete problem.

THEOREM 5.14. *There is a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ in which Σ is a finite set of s-t tgds and such that the following problem is NP-complete: given a source instance I and a number r, is there a target instance $J \in \text{Endom}(I)$ such that* $\text{avg}(R.A)(J) = r$?

PROOF. *(Hint)* The NP-hardness is proved via a reduction from the PARTITION problem; the details are given in the full paper. □

6. Concluding Remarks

We initiated the study of aggregate queries in data exchange by focusing on schema mappings specified by s-t tgds. After examining and rejecting several sets of possible worlds studied earlier for first-order queries, we converged on the set $\text{Endom}(I)$ of all endomorphic images of the canonical universal solution for a source instance I as the "right" set of possible worlds for the semantics of aggregate queries. We then gave polynomial-time algorithms for the range semantics of all scalar aggregate queries with respect to $\text{Endom}(I)$.

The next step is to study the semantics and the complexity of scalar aggregate queries for richer schema mappings. We have already studied the semantics of scalar aggregation queries for schema mappings specified by second-order tgds (SO tgds), which arise as the composition of schema mappings specified by s-t tgds [8]. Since it is known that, for SO tgds, an endomorphic image of $\text{CanSol}(I)$ need not be a solution for I (in fact, $\text{core}(\text{CanSol}(I))$ need not be a solution), we cannot take $\text{Endom}(I)$ as the set of possible worlds. A natural alternative is to take as possible worlds those members of $\text{Endom}(I)$ that are solutions. Using this set of possible worlds, we can show that the tractability results obtained here do not extend to SO tgds. In particular, there are schema mappings specified by SO tgds such that computing the range semantics of count, sum, and avg is NP-hard.

It would be interesting to study aggregate queries for schema mappings specified by s-t tgds and target tgds. Finally, the semantics and the complexity of richer aggregate queries with GROUP BY constructs should also be explored.

7. References

[1] ISO/IEC 9075-2:2003, "SQL/Foundation". ISO/IEC, Section 4.15.4: Aggregate functions.

[2] F. Afrati, C. Li, and V. Pavlaki. Data exchange in the presence of arithmetic comparisons. In *EDBT Conference*, 2008. To appear.

[3] M. Arenas, L. E. Bertossi, J. Chomicki, X. He, V. Raghavan, and J. Spinrad. Scalar aggregation in inconsistent databases. *TCS*, 3(296):405–434, 2003.

[4] M. Arenas and L. Libkin. XML data exchange: consistency and query answering. In *PODS*, pages 13–24, 2005.

[5] R. Fagin. Inverting Schema Mappings. In *PODS*, pages 50–59, 2006.

[6] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data Exchange: Semantics and Query Answering. *TCS*, 336(1):89–124, 2005.

[7] R. Fagin, P. G. Kolaitis, and L. Popa. Data Exchange: Getting to the Core. *ACM TODS*, 30(1):174–210, 2005.

[8] R. Fagin, P. G. Kolaitis, L. Popa, and W.-C. Tan. Composing Schema Mappings: Second-order Dependencies to the Rescue. *ACM TODS*, 30(4):994–1055, 2005.

[9] R. Fagin, P. G. Kolaitis, L. Popa, and W. C. Tan. Quasi-inverses of schema mappings. In *PODS*, pages 123–132, 2007.

[10] A. Fuxman. *Efficient Management of Inconsistent Databases*. PhD thesis, University of Toronto, 2006.

[11] A. Fuxman, E. Fazli, and R. J. Miller. ConQuer: Efficient management of inconsistent databases. In *SIGMOD*, pages 155–166, 2005.

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

[13] G. Gottlob. Computing cores for data exchange: new algorithms and practical solutions. In *PODS*, pages 148–159, 2005.

[14] G. Gottlob and A. Nash. Data exchange: computing cores in polynomial time. In *PODS*, pages 40–49, 2006.

[15] L. M. Haas, M. A. Hernández, H. Ho, L. Popa, and M. Roth. Clio Grows Up: From Research Prototype to Industrial Tool. In *SIGMOD*, pages 805–810, 2005.

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

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

[18] A. Madry. Data exchange: On the complexity of answering queries with inequalities. *Inf. Process. Lett.*, 94(6):253–257, 2005.

[19] S. Melnik. *Generic Model Management: Concepts and Algorithms*, volume 2967 of *Lecture Notes in Computer Science*. Springer, 2004.

[20] A. Nash, P. A. Bernstein, and S. Melnik. Composition of mappings given by embedded dependencies. *ACM TODS*, 32(1):4, 2007.