Answering Aggregate Queries in Data Exchange

Foto Afrati\textsuperscript{1}  Phokion G. Kolaitis\textsuperscript{2}

\textsuperscript{1}National Technical University of Athens

\textsuperscript{2}UC Santa Cruz and IBM Almaden Research Center
Data Exchange

Transform data structured under a schema (source schema) into data structured under another schema (target schema).

Two of the main issues:

- Algorithms for materializing a “good” target instance.
- Semantics and algorithms for answering target queries:

Query Answering

- Earlier work has focused on the certain answers of target FO queries, with emphasis on conjunctive queries.
- In this work we consider aggregate queries over the target:
  1. We give semantics for aggregate query answering.
  2. We give PTIME algorithms for aggregate query answering (data complexity).
Our Framework

Data exchange setting considered:
- source schema;
- target schema;
- source-to-target constraints specified by s-t tgds.

Aggregate queries considered

Scalar aggregation queries

```sql
SELECT f FROM R,
```

where

- `$f$` is one of the aggregate operators $\min(A)$, $\max(A)$, $\text{count}(A)$, $\text{sum}(A)$, $\text{avg}(A)$, and $\text{count}(\ast)$, and

- `$A$` is an attribute of a target relation `$R$`.
Basic Notions (FKMP 2003)

- $\mathcal{M} = (S, T, \Sigma)$ is a schema mapping, where $\Sigma$ is a set of s-t tgds.

- A source-to-target tuple-generating dependency (or an s-t tgd) is a FO-formula $\forall x (\varphi(x) \rightarrow \exists y \psi(x, y))$, where $\varphi(x)$ is a conjunction of atoms over $S$, $\psi(x, y)$ is a conjunction of atoms over $T$, and every variable in $x$ occurs in an atom in $\varphi(x)$.

- Each s-t tgd is a global-and-local-as-view (GLAV) constraint.

- If $I$ a is source instance, then a solution for $I$ under $\mathcal{M}$ is a target instance $J$ such that $(I, J) \models \Sigma$. 
Example

Let $\mathcal{M}$ be specified by the s-t tgd

$$\forall x \forall y (E(x, y) \rightarrow \exists z (F(x, z) \land F(z, y))).$$

If $I = \{E(1, 2)\}$, then the following target instances are solutions for $I$:

- $J_1 = \{E(1, 1), E(1, 2)\}$.
- $J_2 = \{E(1, 2), E(2, 2)\}$.
- $J_3 = \{E(1, w), E(w, 2)\}$, where $w$ is a labeled null.
- $J_4 = \{E(1, w_1), E(w_1, 2), E(1, w_2), E(w_2, 2)\}$, where $w_1, w_2$ are labeled nulls.

There are infinitely many solutions for $I$. 
Definition (FKMP 2003)

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 $h : J \rightarrow J'$.

Note:

- Intuitively, universal solutions are the most general solutions in data exchange; they carry no more and no less information than what is specified by the constraints of the schema mapping.
- Universal solutions are reminiscent of the most general unifiers in logic programming.
- Every two universal solutions are homomorphically equivalent.
Universal Solutions

Example

Let $\mathcal{M}$ be specified by the s-t tgd

$$\forall x \forall y (E(x, y) \rightarrow \exists z (F(x, z) \land F(z, y))).$$

If $I = \{E(1, 2)\}$, then:

- $J_1 = \{E(1, 1), E(1, 2)\}$ is not a universal solution for $I$.
- $J_2 = \{E(1, 2), E(2, 2)\}$ is not a universal solution for $I$.
- $J_3 = \{E(1, w), E(w, 2)\}$ is a universal solution for $I$ (labeled nulls can be mapped to constants)
- $J_4 = \{E(1, w_1), E(w_1, 2), E(1, w_2), E(w_2, 2)\}$ is a universal solution for $I$ (labelled nulls can be mapped to constants or to labelled nulls).
- $J_5 = \{E(1, w), E(w, 2), E(w, w)\}$ is not a universal solution for $I$, even though it contains one.

There are infinitely many universal solutions for $I$. 
Theorem [FKMP 2003]

A canonical universal solution $\operatorname{CanSol}(I)$ for $I$ under $\mathcal{M}$ can be obtained in time polynomial in the size of $I$ using the naive chase procedure.

Naive chase

for every s-t tgd $\varphi(x) \rightarrow \exists y \psi(x, y)$ in $\Sigma$ and for every tuple $a$ from $I$ such that $I \models \varphi(a)$, we introduce a fresh tuple of distinct nulls $u$ and create new facts in the canonical universal solution so that $\psi(a, u)$ holds.
Example

Let $\mathcal{M}$ be specified by the s-t tgd

$$\forall x \forall y (E(x, y) \rightarrow \exists z (F(x, z) \land F(z, y))).$$

If $I = \{E(1, 2)\}$, then the canonical universal solution produced by the naive chase procedure is $J_3 = \{E(1, w), E(w, 2)\}$.

Example

Let $\mathcal{M}'$ be specified by the s-t tgd

$$\forall x \forall y (E(x, y) \rightarrow \exists z_1 \exists z_2 (F(x, z_1) \land F(z_1, y) \land P(z_2))).$$

If $I = \{E(1, 2), E(1, 3)\}$, then the canonical universal solution is

$$J = \{F(1, w_1), F(w_1, 2), P(w_2), F(1, w_3), F(w_3, 3), P(w_4)\}.$$
Cores

**Definition**

A database instance $J'$ is a core of a database instance $J$ if

- $J' \subseteq J$.
- There is a homomorphism $h : J \rightarrow J'$.
- There is no $J^\ast \subset J'$ such that there is a homomorphism $h^\ast : J \rightarrow J^\ast$.

**Example**

- If a graph $G$ is 3-colorable and contains a triangle $K_3$, then $K_3$ is a core of $G$.
- $K_n$ is a core of $K_n$, where $K_n$ is the $n$-clique, $n \geq 2$.
- If $J = \{F(1, w_1), F(w_1, 2), P(w_2), F(1, w_3), F(w_3, 3), P(w_4)\}$, then $J_1$ and $J_2$ are cores of $J$, where
  - $J_1 = \{F(1, w_1), F(w_1, 2), P(w_2), F(1, w_3), F(w_3, 3)\}$.
  - $J_2 = \{F(1, w_1), F(w_1, 2), F(1, w_3), F(w_3, 3), P(w_4)\}$.
Properties of Cores

Facts

- Every (finite) instance has a core.
- All cores of an instance are unique up to isomorphism, hence we can talk about the core of an instance.
- If $J$ and $J'$ are homomorphically equivalent, then their cores are isomorphic.
- Computing the core of an instance is an $\text{NP}$-hard problem.
- (FKP 2003) The following problem is $\text{DP}$-complete: Given two undirected graphs $G$ and $H$, is $H$ the core of $G$?

**Note:** $\text{NP} \cup \text{coNP} \subseteq \text{DP}$. 
The Core of the Universal Solution

Fact:

- Since all universal solutions for an instance $I$ are homomorphically equivalent, they have isomorphic cores.
- Hence, we refer to the core of the universal solutions for $I$.
- The core of the universal solution for $I$ is the smallest universal solution for $I$.

Theorem [FKP 2003]

If $\mathcal{M}$ is a schema mapping specified by s-t tgds, then there is a polynomial-time algorithm such that, given a source instance $I$, it computes the core of the universal solution for $I$. 
Possible Worlds and Certain Answers

Definition

For every instance \( I \) over some schema \( R \), let \( \mathcal{W}(I) \) be a set of instances over some (possibly different) schema \( R^* \) (set of possible worlds). Let \( Q \) be a query over \( R^* \).

- A \( k \)-tuple \( t \) is a certain answer of \( Q \) w.r.t. \( I \) and \( \mathcal{W}(I) \) if for every \( J \in \mathcal{W}(I) \), we have that \( t \in Q(J) \).

\[
\text{certain}(Q, I, \mathcal{W}(I)) = \bigcap_{J \in \mathcal{W}(I)} Q(J).
\]

Note:

- The certain answer semantics is the standard semantics of query answering in the context of incomplete information.

- On the face of the definition, computing the certain answers entails taking an intersection over a potentially infinite set. In general, this is highly non-constructive.
Question:

Fix a schema mapping $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and a FO-query $Q$ over the target $T$. Given a source instance $I$, compute the certain answers of $Q$ w.r.t. $I$. What should the set $\mathcal{W}(I)$ of the set of possible worlds for $I$ be?
Question:
Fix a schema mapping $\mathcal{M} = (S, T, \Sigma)$ and a FO-query $Q$ over the target $T$. Given a source instance $I$, compute the certain answers of $Q$ w.r.t. $I$. What should the set $\mathcal{W}(I)$ of the set of possible worlds for $I$ be?

Three different approaches

1. The set $\text{Sol}(I)$ of all solutions for $I$. [FKMP 2003]
2. The set $\text{USol}(I)$ of all universal solutions for $I$. [FKP 2003]
3. The set $\text{Rep}(\text{CanSol}(I))$ derived from the collection of CWA-solutions for $I$. [Libkin 2006]
Theorem

Fix a schema mapping $\mathcal{M} = (S, T, \Sigma)$ specified by s-t tgds.

- If $Q$ is a union of conjunctive queries over $T$ and $I$ is an $S$-instance, then
  \[ \text{certain}(Q, I, \text{Sol}(I)) = \text{certain}(Q, I, \text{USol}(I)) = \text{certain}(Q, I, \text{Rep}(\text{CanSol}(I))). \]

- If $Q$ is a union of conjunctive queries over $T$, then
  \[ \text{certain}(Q, I, \text{Sol}(I)) = Q(\text{CanSol}(I)) \downarrow. \] Hence, \text{certain}(Q, I, \text{Sol}(I)) is computable in polynomial time. [FKMP 2003]

- If $Q$ is a union of conjunctive queries with inequalities $\neq$ over $T$, then
  \[ \text{certain}(Q, I, \text{USol}(I)) = Q(\text{core}(\text{CanSol}(I))) \downarrow. \] Hence,
  \text{certain}(Q, I, \text{USol}(I)) is computable in polynomial time. [FKP 2003]
Certain Answers of Aggregate Queries


Definition (Q a FO-query, f an aggregate operator)

- 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 $f(Q)(J) = r$.

- $\text{poss}(f(Q), I, \mathcal{W}(I))$ denotes the set of all possible answers of the aggregate query $f(Q)$.

- The aggregate certain answers of the aggregate query $f(Q)$ with respect to $I$ and $\mathcal{W}(I)$ is the interval

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

They are denoted by $\text{agg-certain}(f(Q), I, \mathcal{W}(I))$, 

Bolzano, October 17, 2008
**Definition (informal)**

- An **inconsistent database** is an instance that violates one or more integrity constraints in a given set of constraints.
- A **repair** of an inconsistent database $I$ is an instance $I'$ that satisfies the given constraints and differs from $I$ in a **minimal** way.
- $R(I)$ is the set of all repairs of $I$. 
Definition (informal)

- An inconsistent database is an instance that violates one or more integrity constraints in a given set of constraints.
- A repair of an inconsistent database $I$ is an instance $I'$ that satisfies the given constraints and differs from $I$ in a minimal way.
- $\mathcal{R}(I)$ is the set of all repairs of $I$.

Theorem [Arenas et al. - 2003]

Computing $\text{agg-certain}(\text{avg}(R.A), I, \mathcal{R}(I))$ can be coNP-hard even if the set of integrity constraints consists of just two functional dependencies.
Approach:
We will adopt the aggregate certain answers as the semantics of aggregate target queries in data exchange.

Question:
What is the right choice of possible worlds in this case?
Semantics of Aggregate Queries in Data Exchange

**Approach:**

We will adopt the **aggregate certain answers** as the semantics of aggregate target queries in data exchange.

**Question:**

What is the *right* choice of possible worlds in this case?

**Sets of possible worlds for FO-queries in data exchange:**

- The set $\text{Sol}(I)$ of all solutions (FKMP03).
- The set $\text{USol}(I)$ of all *universal* solutions (FKP03).
- The set $\text{Rep}(\text{CanSol}(I))$ obtained from CWA solutions (Libkin 2006).

**Fact:**

Each of these sets of possible worlds gives rise to rather *trivial* aggregate certain answers.
### Sol(I) and USol(I) as Sets of Possible Worlds

#### Fact (Using Sol(I) as $\mathcal{W}(I)$)

If $I$ is a source instance and $f$ is one of min, max, sum, avg, then

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

#### Fact (Using USol(I) as $\mathcal{W}(I)$)

Let $a = \text{min}(R.A)(\text{CanSol}(I))$ and $b = \text{max}(R.A)(\text{CanSol}(I))$

1. $\text{agg-certain}(\text{min}(R.A), I, \text{USol}(I)) = a.$
2. $\text{agg-certain}(\text{max}(R.A), I, \text{USol}(I)) = b.$
3. If $a = b$, then $\text{agg-certain}(\text{avg}(R.A), I, \text{USol}(I)) = a.$
4. If $a < b$, then $\text{agg-certain}(\text{avg}(R.A), I, \text{USol}(I)) = (a, b).$
Definition

Let $\mathcal{M} = (\mathbf{ST}, \Sigma)$ be a schema mapping specified by s-t tgds. Libkin (2006) defined the concept of a CWA-solution for a source instance $I$ by giving a set of “axioms” that such a solution should satisfy.

Theorem [Libkin06]

The following two statements are equivalent.

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 $\text{CanSol}(I)$.
Rep(\text{CanSol}(I)) as Sets of Possible Worlds

**Definition**

- \text{Rep}(J) coincides with the set of null-free homomorphic images of \text{J}.
- Libkin took the set \bigcup_{J \in \text{CWA}(I)} \text{Rep}(J) as the set of possible worlds for the semantics of FO-queries in data exchange.

**Proposition**

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

In words, the set of possible worlds \mathcal{W}(I) considered by Libkin is simply the set of all null-free homomorphic images of \text{CanSol}(I).
Rep(\text{CanSol}(I)) as Sets of Possible Worlds

**Definition**
- \text{Rep}(J) coincides with the set of null-free homomorphic images of \text{J}.
- Libkin took the set \( \bigcup_{J \in \text{CW}(I)} \text{Rep}(J) \) as the set of possible worlds for the semantics of FO-queries in data exchange.

**Proposition**
\( \bigcup_{J \in \text{CW}(I)} \text{Rep}(J) = \text{Rep}(\text{CanSol}(I)). \)
In words, the set of possible worlds \( \mathcal{W}(I) \) considered by Libkin is simply the set of all null-free homomorphic images of \text{CanSol}(I).

**Fact (Using \text{Rep}(\text{CanSol}(I)) as \mathcal{W}(I))**
If \text{CanSol}(I) contains at least one fact \( R(t) \) in which \( t[A] \) is a null, then
\( \text{agg-certain}(f(R), I, \text{Rep}(\text{CanSol}(I))) = (-\infty, \infty). \)
Endomorphic Images of $\text{CanSol}(I)$

Notation

If $I$ is a source instance, then $\text{Endom}(I)$ stands for the set of all endomorphic images of $\text{CanSol}(I)$. 
### Notation

If $I$ is a source instance, then $\text{Endom}(I)$ stands for the set of all endomorphic images of $\text{CanSol}(I)$.

### Example

Let $\mathcal{M}'$ be specified by the s-t tgd

$$\forall x \forall y (E(x, y) \rightarrow \exists z_1 \exists z_2 (F(x, z_1) \land F(z_1, y) \land P(z_2))).$$

If $I = \{E(1, 2), E(1, 3)\}$, then $\text{Endom}(I)$ consists of

$$J = \{F(1, w_1), F(w_1, 2), P(w_2), F(1, w_3), F(w_3, 3), P(w_4)\}$$

$$J_1 = \{F(1, w_1), F(w_1, 2), P(w_2), F(1, w_3), F(w_3, 3)\}$$

$$J_2 = \{F(1, w_1), F(w_1, 2), F(1, w_3), F(w_3, 3), P(w_4)\}.$$

Proposal

Use $\text{Endom}(I)$ as sets of possible worlds $\mathcal{W}(I)$ for the semantics of aggregate queries in data exchange.

Properties

- $\text{Endom}(I)$ contains both $\text{CanSol}(I)$ and $\text{core}(\text{CanSol}(I))$ as members. Moreover, $\text{Endom}(I) \subseteq \text{USol}(I)$.
- Every member of $\text{Endom}(I)$ is a sub-instance of $\text{CanSol}(I)$; the converse, however, need not hold.
- Every member of $\text{Endom}(I)$ is a CWA-solution for $I$; the converse, however, need not hold.
Endom($I$) as Sets $\mathcal{W}(I)$ of Possible Worlds

Some reasons for this choice:

- The members of Endom($I$) adhere to a **strict** closed world assumption.
- If Endom($I$) are used as sets of possible worlds for the semantics of conjunctive queries $Q$, then
  \[ \text{certain}(Q, I, \text{Endom}(I)) = \text{certain}(Q, I, \text{Sol}(I)). \]
- agg-certain($f(Q), I, \text{Endom}(I)$) is **non-trivial** semantics for aggregate queries $f(Q)$. 
### Proposition

CanSol(I) and core(CanSol(I)) suffice for max, min, count, and a special case of sum.

- For every instance $T \in \text{Endom}(I)$, we have that $\max(R.A)(T) = \max(R.A)(\text{CanSol}(I)) = a$. Similarly for min.
- $\text{agg-certain(count}(R.A), I, \text{Endom}(I)) = [\text{count}(R.A)(\text{core}(\text{CanSol}(I))), \text{count}(R.A)(\text{CanSol}(I))].$
- If all numeric constants in $I$ are non-negative integers, then $\text{agg-certain(sum}(R.A), I, \text{Endom}(I)) = [\text{sum}(R.A)(\text{core}(\text{CanSol}(I))), \text{sum}(Q)(\text{CanSol}(I))].$

### Note

For sum in the general case, we use a simpler version of the technique that we will use for the average.
Example

- Schema mapping $\mathcal{M}$ consisting of
  \[
  \forall x, y (P(x, y) \rightarrow T(x, y)) \\
  \forall x, y (Q(x, y) \rightarrow \exists z T(x, z)).
  \]

- Source instance
  \[I_n = \{P(a_1, b_1), \ldots, P(a_n, b_n), Q(a_1, c_1), \ldots, Q(a_n, c_n)\}.
  \]

- CanSol($I_n$) is
  \[J_n = \{T(a_1, b_1), \ldots, T(a_n, b_n), T(a_1, u_1), \ldots, T(a_n, u_n)\}.
  \]

- Every subset $K$ of $\{1, \ldots, n\}$ determines an endomorphism $h_K$ of $J_n$, and vice versa.

- Thus, Endom($I$) consists of exponentially many endomorphic images, one for each subset of $\{1, \ldots, n\}$.
Example (Continued)

- $K \subseteq \{1, \ldots, n\}$.
- $\text{count}((T.A)^J_K) = n + |K|$ and $\text{sum}((T.A)^J_K) = (\sum_{i=1}^n a_i) + (\sum_{i \in K} a_i)$.
- Consequently,
  
  $\text{agg-certain}(\text{count}(T.A), I_n, \text{Endom}(I_n)) = [n, 2n]$ and
  $\text{agg-certain}(\text{sum}(T.A), I_n, \text{Endom}(I_n)) = [\sum_{i=1}^n a_i, 2 \sum_{i=1}^n a_i]$.
- Moreover, the endpoints of these intervals are obtained by evaluating $\text{count}(T.A)$ and $\text{sum}(T.A)$ on $\text{core}(\text{CanSol}(I_n))$ and on $\text{CanSol}(I_n)$.
Answering queries with the *average*, however, is more complicated. Take the source instance

\[ I = \{(1, b_1), (2, b_2), (3, b_3)\}. \]

Then

- \( \text{agg-certain}(\text{avg}(T.A), I, \text{Endom}(I)) = [7/4, 9/4] \).
- \( \text{avg}(T.A)(\text{core}(\text{CanSol}(I))) = 2 = \text{avg}(T.A)(\text{CanSol}(I)). \)
Theorem

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

Proof Hint:

Will only describe some of the concepts and the ingredients for the algorithm.
Blocks and Block Homomorphisms

Definition (FKP 2003)

Let \( K \) be a target instance.

- The Gaifman graph of the nulls of \( K \) has the nulls of \( K \) as nodes; two nulls are connected via an edge if they occur in some fact of \( K \).
- A block of \( K \) is a connected component of the Gaifman graph of \( K \).
- A block homomorphism of \( B \) is a homomorphism from \( B \) to \( K \).

Fact

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

- We design a PTIME algorithm for \( \text{avg} \) that, given \( I \), finds endomorphic images \( J \) and \( J' \) of \( \text{CanSol}(I) \) that realize the optimum (minimum and maximum) values for \( \text{avg} \).
- We can partition the set of integers in polynomially many critical intervals determined by the blocks.
- For each critical interval, we can decide which block homomorphism is optimum, supposing that the value of the optimum \( \text{avg} \) is in this interval.
- We can find the optimum endomorphic image by assembling the optimum block homomorphisms.
- Assembling block homomorphisms requires care.
Example

- Revisit $\mathcal{M}$ consisting of

\[
\forall x, y (P(x, y) \rightarrow T(x, y)) \\
\forall x, y (Q(x, y) \rightarrow \exists z T(x, z)).
\]

- For every source instance $I$, each block of $\text{CanSol}(I)$ is of size one.
- Critical intervals are determined by the values of the attribute $A$.
- The problem of finding an endomorphic image with the minimum average is literally equivalent to the following combinatorial problem:
  Given a bag $S$ of positive integers, find a sub-bag $S'$ of $S$ such that:
  (a) $S$ and $S'$ have the same set of distinct numbers; and
  (b) the average of the members of $S'$ is minimized.
- Thus, computing $\text{agg-certain}(\text{avg}(T.A), I, \text{Endom}(I))$ is an algorithmically interesting problem, even for seemingly very simple schema mappings $\mathcal{M}$. 
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**

There is a schema mapping $\mathcal{M} = (S, T, \Sigma)$ in which $\Sigma$ 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 $\text{avg}(R.A)(J) = r$?

**Hint of Proof:**

Reduction from the **Partition Problem**.
Concluding Remarks

Summary of Contributions

- We have given semantics for aggregate queries in data exchange.
- We have given polynomial algorithms to compute the aggregate certain answers under these semantics and for schema mappings specified by s-t tgds.
- More recently, we have shown that computing the aggregate certain answers for schema mappings specified by SO tgds is NP-hard.

Next Steps

- Study aggregate queries for schema mappings specified by s-t tgds and target tgds.
- Semantics and the complexity of richer aggregate queries with GROUP BY constructs.