Answering Aggregate Queries in Data Exchange

Foto Afrati[1] Phokion G. Kolaitis[2]

[1]National Technical University of Athens

[2]UC Santa Cruz and IBM Almaden Research Center

Data Exchange

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

$$\texttt{SELECT}\ f\ \texttt{FROM}\ R,$$

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
- A is an attribute of a target relation R.

Schema Mappings and Data Exchange

Basic Notions (FKMP 2003)

- $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ is a schema mapping, where Σ is a set of s-t tgds.
- A source-to-target tuple-generating dependency (or an s-t tgd) is a FO-formula $\forall \mathbf{x}(\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y}))$, 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 $\varphi(\mathbf{x})$.
- 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 $(I, J) \models \Sigma$.

Schema Mappings and Data Exchange

Example

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

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

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.

Universal Solutions

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 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

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

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.

Canonical Universal Solutions and the Chase Procedure

Theorem [FKMP 2003]

A canonical universal solution $\mathrm{CanSol}(I)$ 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 $\varphi(\mathbf{x}) \rightarrow \exists \mathbf{y} \psi(\mathbf{x}, \mathbf{y})$ in Σ and for every tuple $\mathbf{a}$ from I such that $I \models \varphi(\mathbf{a})$, 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.

Canonical Universal Solutions and the Chase Procedure

Example

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

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

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

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

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 $J^* \subset J'$ such that there is a homomorphism $h^* : J \rightarrow J^*$.

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 NP-hard problem.
- (FKP 2003) The following problem is DP-complete: Given two undirected graphs G and H, is H the core of G?

  Note: $\mathrm{NP} \cup \mathrm{coNP} \subseteq \mathrm{DP}$.

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

- A k-tuple $\mathbf{t}$ 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 $\mathbf{t} \in Q(J)$.
- $\mathrm{certain}(Q, I, \mathcal{W}(I)) = \bigcap_{J \in \mathcal{W}(I)} Q(J)$.

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.

Certain Answers of FO-Queries in Data Exchange

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?

Certain Answers of FO-Queries in Data Exchange

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?

Three different approaches

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

Certain Answers of FO-Queries in Data Exchange

Theorem

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

- If Q is a union of conjunctive queries over $\mathbf{T}$ and I is an $\mathbf{S}$-instance, then $\mathrm{certain}(Q, I, \mathrm{Sol}(I)) = \mathrm{certain}(Q, I, \mathrm{USol}(I)) = \mathrm{certain}(Q, I, \mathrm{Rep}(\mathrm{CanSol}(I)))$.
- If Q is a union of conjunctive queries over $\mathbf{T}$, then $\mathrm{certain}(Q, I, \mathrm{Sol}(I)) = Q(\mathrm{CanSol}(I)) \downarrow$. Hence, $\mathrm{certain}(Q, I, \mathrm{Sol}(I))$ is computable in polynomial time. [FKMP 2003]
- If Q is a union of conjunctive queries with inequalities $\neq$ over $\mathbf{T}$, then $\mathrm{certain}(Q, I, \mathrm{USol}(I)) = Q(\mathrm{core}(\mathrm{CanSol}(I))) \downarrow$. Hence, $\mathrm{certain}(Q, I, \mathrm{USol}(I))$ is computable in polynomial time. [FKP 2003]

Certain Answers of Aggregate Queries

M. Arenas, L. E. Bertossi, J. Chomicki, X. He, V. Raghavan, and J. Spinrad: Scalar aggregation in inconsistent databases - 2003.

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 $f(Q)(J) = r$.
- $\mathrm{poss}(f(Q), I, \mathcal{W}(I))$ 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

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

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

Aggregate Query Answering in Inconsistent Databases

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.

Aggregate Query Answering in Inconsistent Databases

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 $\mathrm{agg\text{-}certain}(avg(R.A), I, \mathcal{R}(I))$ can be coNP-hard even if the set of integrity constraints consists of just two functional dependencies.

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?

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

Fact:

Each of these sets of possible worlds gives rise to rather trivial aggregate certain answers.

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

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

If I is a source instance and f is one of min, max, sum, avg, then
$\mathrm{agg\text{-}certain}(f(R), I, \mathrm{Sol}(I)) = (-\infty, \infty)$.

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

.

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

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

CWA-Solutions and Possible Worlds

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 $\mathrm{CanSol}(I)$; moreover, there is a homomorphism from J to $\mathrm{CanSol}(I)$.

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

Definition

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

Proposition

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

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

Definition

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

Proposition

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

Fact (Using $\mathrm{Rep}(\mathrm{CanSol}(I))$ as $\mathcal{W}(I)$)

If $\mathrm{CanSol}(I)$ contains at least one fact $R(\mathbf{t})$ in which $\mathbf{t}[A]$ is a null, then

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

Endomorphic Images of $\mathrm{CanSol}(I)$

Notation

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

Endomorphic Images of $\mathrm{CanSol}(I)$

Notation

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

Example

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

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

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

$$\begin{aligned} J &= \{F(1,w_1), F(w_1,2), P(w_2), F(1,w_3), F(w_3,3), P(w_4)\} \\ J_1 &= \{F(1,w_1), F(w_1,2), P(w_2), F(1,w_3), F(w_3,3)\} \\ J_2 &= \{F(1,w_1), F(w_1,2), F(1,w_3), F(w_3,3), P(w_4)\}. \end{aligned}$$

Endomorphic Images of $\mathrm{CanSol}(I)$

Proposal

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

Properties

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

$\mathrm{Endom}(I)$ as Sets $\mathcal{W}(I)$ of Possible Worlds

Some reasons for this choice:

- The members of $\mathrm{Endom}(I)$ adhere to a strict closed world assumption.
- If $\mathrm{Endom}(I)$ are used as sets of possible worlds for the semantics of conjunctive queries Q, then

$$\mathrm{certain}(Q, I, \mathrm{Endom}(I)) \;=\; \mathrm{certain}(Q, I, \mathrm{Sol}(I)).$$

- $\text{agg-certain}(f(Q), I, \mathrm{Endom}(I))$ is non-trivial semantics for aggregate queries $f(Q)$.

PTIME Algorithms for max, min, count

Proposition

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

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

Note

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

Exponentially Many Endomorphic Images

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
  $I_n = \{P(a_1, b_1), \ldots, P(a_n, b_n), Q(a_1, c_1), \ldots, Q(a_n, c_n)\}$.
- $\mathrm{CanSol}(I_n)$ 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, $\mathrm{Endom}(I)$ consists of exponentially many endomorphic images, one for each subset of $\{1, \ldots, n\}$.

Non-trivial Semantics for Aggregate Queries

Example (Continued)

- $K \subseteq \{1, \ldots, n\}$.
- $\mathrm{count}((T.A)^{J_K}) = n + |K|$ 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
$$\text{agg-certain}(\mathrm{sum}(T.A), I_n, \mathrm{Endom}(I_n)) = [\sum_{i=1}^{n} a_i, 2\sum_{i=1}^{n} a_i].$$
- 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 $\mathrm{CanSol}(I_n)$.

Non-trivial Semantics for Aggregate Queries

Example (Continued)

Answering queries with the average, however, is more complicated. Take the source instance

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

Then

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

PTIME Algorithm for avg

Theorem

Let $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a schema mapping in which Σ 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 $\mathrm{agg\text{-}certain}(\mathrm{avg}(R.A), I, \mathrm{Endom}(I))$.

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 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 $\mathrm{CanSol}(I)$ is bounded by $p(|I|)$.
- 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 constant c.

PTIME Algorithm for avg

Basic Ingredients

- We design a PTIME algorithm for avg that, given I, finds endomorphic images J and J' of $\mathrm{CanSol}(I)$ that realize the optimum (minimum and maximum) values for 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 avg is in this interval.
- We can find the optimum endomorphic image by assembling the optimum block homomorphisms.
- Assembling block homomorphisms requires care.

PTIME Algorithm for avg

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 $\mathrm{CanSol}(I)$ 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 $\text{agg-certain}(\mathrm{avg}(T.A), I, \mathrm{Endom}(I))$ is an algorithmically interesting problem, even for seemingly very simple schema mappings $\mathcal{M}$.

Intractability of Aggregate Possible Answers

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 $\mathcal{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ 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 $J \in \mathrm{Endom}(I)$ such that $\mathrm{avg}(R.A)(J) = r$?

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.