Foundations of Information Integration under Bag Semantics

André Hernich
University of Liverpool

Phokion G. Kolaitis
UC Santa Cruz & IBM Research - Almaden
Logic and Information Integration

• Two uses of logic in databases:
  – Logic as a database query language
  – Logic as a specification language to express integrity constraints

• Both uses occur in the formalization and analysis of information integration

• So far, information integration has been studied under set semantics

• This work aims to study information integration under bag semantics.
The Relational Database Model

Introduced by E.F. Codd in 1969

• Relational Database
  \[ D = (R_1, \ldots, R_m), \text{ where} \]
  – each \( R_i \) is a relation of a specified arity with named attributes.
  – EMPLOYEE (name, department, salary)

• First-Order Logic used as a database query language.

• First-Order Logic forms the core of SQL, the main commercial database query language.
Conjunctive Queries

**Definition:** A conjunctive query is a query expressible by a FO-formula built from atomic formulas, \( \land \), and \( \exists \)

\[
\{ (x_1, \ldots, x_k): \exists z_1 \ldots \exists z_m \chi(x_1, \ldots, x_k, z_1, \ldots, z_k) \},
\]

where \( \chi(x_1, \ldots, x_k, z_1, \ldots, z_k) \) is a conjunction of atomic formulas \( R_i (y_1, \ldots, y_m) \).

**Fact:**

- Conjunctive queries are expressed using the `SELECT ... FROM ... WHERE` construct of SQL.
- Conjunctive queries are among the most frequently asked database queries.
Examples of Conjunctive Queries

– **Salaries of employees** (Unary query)
  \[
  \{ s \mid \exists n \exists d \text{ EMPLOYEE}(n,d,s) \}
  \]

– **Path of Length 2:** (Binary query)
  \[
  \{ (x,y) \mid \exists z \ (\text{E}(x,z) \land \text{E}(z,y)) \}
  \]

– **Existence of a triangle:** (Boolean query)
  \[
  \exists x \exists y \exists z \ (\text{E}(x,y) \land \text{E}(y,z) \land \text{E}(z,x))
  \]
Set Semantics of Conjunctive Queries

– **Salaries of employees** (Unary query)
  \[
  \{ s \mid \exists n \exists d \text{EMPLOYEE}(n,d,s) \}
  \]
  Returns the set of all distinct salaries of employees.

– **Path of Length 2:** (Binary query)
  \[
  \{ (x,y) \mid \exists z (E(x,z) \land E(z,y)) \}
  \]
  Returns the set of all pairs \((a,b)\) connected via a path of length 2.

– **Existence of a triangle:** (Boolean query)
  \[
  \exists x \exists y \exists z (E(x,y) \land E(y,z) \land E(z,x))
  \]
  Tells whether or not the graph contains a triangle.
Bag Semantics of Conjunctive Queries

Fact: SQL uses bag (multiset) semantics (unless explicitly told otherwise via the SELECT DISTINCT construct).

- **Salaries of employees** (Unary query)
  \[ \{ s \mid \exists n \exists d \text{ EMPLOYEE}(n,d,s) \} \]
  \[ \{ (s:m) \mid \text{there are m employees earning salary s} \} \]

- **Path of Length 2:** (Binary query)
  \[ \{ (x,y) \mid \exists z (E(x,z) \land E(z,y)) \} \]
  \[ \{ (a,b:m) \mid \text{there are m paths of length 2 between a and b} \} \]

- **Existence of a triangle:** (Boolean query)
  \[ \exists x \exists y \exists z (E(x,y) \land E(y,z) \land E(z,x)) \]
  \[ 6 \cdot \# \text{ of triangles in E} \]
Set Semantics vs. Bag Semantics

Fact:
• The algorithmic properties of conjunctive queries under set semantics are well understood.
• The algorithmic properties of conjunctive queries under bag semantics are **not** well understood.

Conjunctive Query Containment (CQC)
• Given two conjunctive queries \( q_1 \) and \( q_2 \) of the same arity, is it true that \( q_1 \subseteq q_2 \)? (i.e., \( q_1(D) \subseteq q_2(D) \), for every \( D \))

Fact:
• Under set semantics, CQC is **NP-complete**.
• Under bag semantics, it is **not** known whether or not QCQ is decidable.
Information Integration

• Data may reside
  – at several different sites
  – in several different formats.

• Applications need to access, process, and query these data.

• Data Exchange:
  – A fundamental problem in information integration
  – Described as the “oldest problem in databases”
  – Formalized and studied in depth in the past 15 years.
Data Exchange

• Transform data structured under a source schema into data structured under a different target schema.
• Answer queries over the target schema.
Schema Mappings and Data Exchange

- **Schema Mapping** $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$
  - **Source schema** $\mathbf{S}$, **Target schema** $\mathbf{T}$
  - $\Sigma$: High-level, declarative assertions that specify the relationship between $\mathbf{S}$ and $\mathbf{T}$.
- Let $\mathbf{I}$ be a source instance. A **solution** for $\mathbf{I}$ w.r.t. $\mathbf{M}$ is a target instance $\mathbf{J}$ such that $(\mathbf{I},\mathbf{J}) \models \Sigma$
- The **certain answers** of a target query $\mathbf{q}$ on $\mathbf{I}$ w.r.t. $\mathbf{M}$
  \[
  \text{certain}(\mathbf{q},\mathbf{I},\mathbf{M}) = \bigcap \{ \mathbf{q}(\mathbf{J}) \mid \mathbf{J} \text{ is a solution for } \mathbf{I} \text{ w.r.t. } \mathbf{M} \}
  \]
Question:
What is a “good” schema-mapping specification language?

Fact:
Unrestricted use of FO leads to undecidability (e.g., undecidability of certain answers of conjunctive queries ).

Answer:
The language of GLAV (global-and-local as view) constraints strikes a good balance between expressive power and good algorithmic properties.
GLAV Constraints and GLAV Mappings

Definition: \( S \) source schema, \( T \) target schema.

- **GLAV constraint:** a FO-sentence of the form
  \[ \forall x \left( q_1(x) \rightarrow q_2(x) \right) \], where
  \( q_1(x) \) is a conjunctive query over \( S \) and \( q_2(x) \) is a conjunctive query over \( T \).

- **GLAV mapping:** A schema mapping \( M = (S, T, \Sigma) \) such that \( \Sigma \) is a finite set of GLAV constraints.

- **GAV constraint:** a GLAV constraint in which \( q_2(x) \) is a single atom over \( T \).

- **GAV mapping:** A schema mapping \( M = (S, T, \Sigma) \) such that \( \Sigma \) is a finite set of GAV constraints.
Expressive Power of GLAV Constraints

- **Copy (Nicknaming):**
  - \( \forall x_1 \cdots \forall x_n (P(x_1,\ldots,x_n) \rightarrow R(x_1,\ldots,x_n)) \) (GAV constraint)

- **Projection:**
  - \( \forall x \forall y \forall z (P(x,y,z) \rightarrow R(x,y)) \) (GAV constraint)

- **Column Augmentation:**
  - \( \forall x \forall y (P(x,y) \rightarrow \exists z \ R(x,y,z)) \)

- **Decomposition:**
  - \( \forall x \forall y \forall z (P(x,y,z) \rightarrow R(x,y) \land T(y,z)) \)

- **Join:**
  - \( \forall x \forall y \forall z (E(x,z) \land F(z,y) \rightarrow R(x,y,z)) \) (GAV constraint)

- **Combinations of the above** (“join + column augmentation + ...”)
  - \( \forall x \forall y \forall z (E(x,z) \land F(z,y) \rightarrow \exists w (R(x,y) \land T(x,y,z,w))) \)
Algorithmic Properties of GLAV Mappings

Theorem (Fagin, K ..., Miller, Popa – 2005)

Let $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ be a GLAV mapping.

• Let $q$ be a conjunctive query over the target schema $\mathbf{T}$. There is a PTIME-algorithm that, given a source instance $\mathbf{I}$, computes the certain answers $\text{certain}(q, \mathbf{I}, \mathbf{M})$.

• There a PTIME-algorithm that, given a source instance $\mathbf{I}$, computes a universal solution $\mathbf{J}$ for $\mathbf{I}$ (i.e., a “most general” solution for $\mathbf{I}$ w.r.t. $\mathbf{M}$).
Bag Semantics for Schema Mappings

• So far, the investigation of data exchange and schema mappings has been carried out under set semantics.

• The goal of the present work is to investigate data exchange and schema mappings under bag semantics.

• Conceptual Contributions:
  – Bag semantics for GLAV constraints.
  – Two different bag semantics for GLAV mappings.

• Technical Contributions:
  – Complexity-theoretic analysis of the certain answers of conjunctive queries under bag semantics.
Bag Semantics for GLAV Constraints

Definition: GLAV constraint \( \forall x \ (q_1(x) \rightarrow q_2(x)) \).

Let I be a bag source instance and J be a bag target instance. Then \((I,J)\) satisfies \( \forall x \ (q_1(x) \rightarrow q_2(x)) \) if \( q_1(I) \subseteq_{\text{BAG}} q_2(J) \).

Examples:

- \((I,J)\) satisfies \( \forall x \ (P(x) \rightarrow R(x)) \) means that, for every \( a \) in \( P \), multiplicity of \( a \) in \( P \) is \( \leq \) multiplicity of \( a \) in \( R \).
- Let \( \psi \) be \( \forall x \ (\exists y \ P(x,y) \rightarrow R(x)) \)
  - If \( I = \{ P(a,b:2), P(a,c:3) \} \), \( J = \{ R(a:5) \} \), then \((I,J)\) satisfies \( \psi \).
  - If \( I = \{ P(a,b:2), P(a,c:3) \} \), \( J = \{ R(a:4) \} \), then \((I,J)\) does not satisfy \( \psi \).
Bag Semantics for GLAV Mappings

Motivation: GLAV mapping $M = (S, T, \Sigma)$, where $\Sigma$ consists of
$\forall x (P(x) \rightarrow R(x))$ and $\forall x (Q(x) \rightarrow R(x))$.

- Intuitively, $(I,J)$ satisfies $\Sigma$ is $R$ contains the union of $P$ and $Q$.
- However, there are two notions of union of bags $B_1$ and $B_2$.

- Max-Union $B_1 \cup B_2$: the multiplicity of a tuple a in $B_1 \cup B_2$ is the maximum of the multiplicities of a in $B_1$ and $B_2$.
- Sum-Union $B_1 \uplus B_2$: the multiplicity of a tuple a in $B_1 \uplus B_2$ is the sum of the multiplicities of a in $B_1$ and $B_2$.

Note: SQL supports Sum-Union via the UNION ALL construct.
Bag Semantics for GLAV Mappings

Definition: GLAV mapping $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$

- $J$ is an incognizant solution (i-solution) for $I$ w.r.t. $M$ if $(I,J)$ satisfies every constraint $\psi$ in $\Sigma$.

- $J$ is a cognizant solution (c-solution) for $I$ w.r.t. $M$ if for every constraint $\psi$ in $\Sigma$, there is a target instance $J_\psi$ such that $(I,J_\psi)$ satisfies $\psi$ and $\bigcup J_\psi \subseteq J$.

Note:

- i-solutions generalize max-union.
- c-solutions generalize sum-union.
- Every c-solution is an i-solution.
- An i-solution need not be a c-solution.
Bag Semantics for Certain Answers

Definition: GLAV mapping $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, $q$ conjunctive query over the target schema $\mathbf{T}$, and $I$ a source instance.

- $i$-certain($q$, $I$, $\mathbf{M}$) = $\bigcap \{q(J): J$ is an $i$-solution for $I$ w.r.t. $\mathbf{M}\}$.
- $c$-certain($q$, $I$, $\mathbf{M}$) = $\bigcap \{q(J): J$ is a $c$-solution for $I$ w.r.t. $\mathbf{M}\}$.

Note: The intersection $\bigcap$ of bags returns the minimum of the multiplicities of tuples in the intersecting sets.

Decision Problems for Boolean conjunctive queries

- $i$-QA($\mathbf{M}$, $q$): Given a source instance $I$ and some $m \geq 1$, is $i$-certain($q$, $I$, $\mathbf{M}$) $\geq m$?
- $c$-QA($\mathbf{M}$, $q$): Given a source instance $I$ and some $m \geq 1$, is $c$-certain($q$, $I$, $\mathbf{M}$) $\geq m$?
Complexity of Certain Answers

Theorem:

• If $M = (S, T, \Sigma)$ is a GLAV mapping and $q$ is a Boolean conjunctive query, then $i$-$QA(M,q)$ and $c$-$QA(M,q)$ are in coNP.

• There are GLAV mappings $M$ and Boolean conjunctive queries $q$ such that $i$-$QA(M,q)$ and $c$-$QA(M,q)$ are coNP-complete.

• If $M = (S, T, \Sigma)$ is a GAV mapping and $q$ is a Boolean conjunctive query, then $i$-$QA(M,q)$ and $c$-$QA(M,q)$ are in PTIME.
Minimal Extensions of GAV Constraints

Definition: GLAV constraint $\forall \mathbf{x} (q_1(\mathbf{x}) \rightarrow q_2(\mathbf{x}))$

- **GAV constraint:** $q_2(\mathbf{x})$ is a single atom
- **Elementary constraint:** $q_2(\mathbf{x})$ is a single atom or an existentially quantified single atom.
- **Full constraint:** $q_2(\mathbf{x})$ is a conjunction of atoms (no $\exists$)

Examples:

- **Projection:** GAV constraint
  $\forall x \forall y \forall z (P(x,y,z) \rightarrow R(x,y))$
- **Column Augmentation:** Elementary constraint
  $\forall x \forall y (P(x,y) \rightarrow \exists z R(x,y,z))$
- **Decomposition:** Full Constraint
  $\forall x \forall y \forall z (P(x,y,z) \rightarrow R(x,y) \land T(y,z))$
Complexity of Certain Answers

Theorem:
• If $M = (S, T, \Sigma)$ is an elementary mapping and $q$ is a Boolean conjunctive query, then $c$-QA$(M,q)$ is in PTIME. Moreover, every source instance has a $c$-universal solution.

• There is an elementary mapping $M$ and a Boolean conjunctive query $q$ such that $i$-QA$(M,q)$ is coNP-complete.

• There is a full mapping $M$ and a Boolean conjunctive query $q$ such that $i$-QA$(M,q)$ and $c$-QA$(M,q)$ are coNP-complete.

Note: Under set semantics, every full mapping is logically equivalent to a GAV mapping.
Synopsis and Outlook

- Studied query answering in data exchange under bag semantics
- Introduced two flavors of bag semantics: **incognizant** and **cognizant**
- Studied the complexity of certain answers under bag semantics

<table>
<thead>
<tr>
<th>Type of Mapping</th>
<th>i-certain answers</th>
<th>c-certain answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAV</td>
<td>PTIME</td>
<td>PTIME</td>
</tr>
<tr>
<td>Elementary</td>
<td>coNP-complete</td>
<td>PTIME</td>
</tr>
<tr>
<td>Full</td>
<td>coNP-complete</td>
<td>coNP-complete</td>
</tr>
</tbody>
</table>

- Investigate approximation algorithms for i-certain and c-certain
- Investigate **ETL** (Extract-Transform-Load) tools under bag semantics
  - Most ETL transformations are specified by elementary mappings
- Nikolaou et al. studied bag semantics of **ontology-based data access**
  - Data integration with constraints expressible in **description logics**
  - Considered i-certain answers only
BACK-UP SLIDES
Complexity of Certain Answers

**Theorem:** There is a full mapping $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$ and a Boolean conjunctive query $q$ such that $i$-$\text{QA}(\mathbf{M}, q)$ and $c$-$\text{QA}(\mathbf{M}, q)$ are coNP-complete.

**Proof:** Reduction from \textsc{Positive Not-All-Equal 3Sat} (a.k.a., \textsc{3-Hypergraph 2-Colorability})

- $\mathbf{M} = (\mathbf{S}, \mathbf{T}, \Sigma)$, where $\Sigma$ consists of
  - $\forall x \; \forall t \; \forall f \; (V(x,t,f) \rightarrow A(x,t) \land A(x,f))$
  - $\forall x \; \forall y \; \forall z \; (C(x,y,z) \rightarrow C'(x,y,z))$.
- $q: \exists x \; \exists y \; \exists z \; \exists v \; (C'(x,y,z) \land A(x,v) \land A(y,v) \land A(z,v))$. 
Complexity of Certain Answers

**Theorem:** There is an elementary mapping $M = (S, T, \Sigma)$ and a Boolean conjunctive query $q$ such that $i$-$QA(M, q)$ is coNP-complete.

**Proof:** Reduction from **POSITIVE NOT-ALL-EQUAL 3SAT**

- **$M = (S, T, \Sigma)$**, where $\Sigma$ consists of
  - $\forall x (P(x) \rightarrow \exists y T'(x,x,y))$
  - $\forall x (P(x) \rightarrow \exists z T'(x,z,x))$
  - $\forall x \forall y \forall z (W(x,y,z) \rightarrow W'(x,y,z))$, where $W \in \{R, S_t, S_f, C, T\}$.
- **$q$:** $\exists x \exists y \exists z \exists v (C'(x,y,z) \land \theta(x,v) \land \theta(y,v) \land \theta(z,v))$. 