Principles of Managing Uncertain Data

Lecture 8: Consistent Query Answering

Acknowledgment

Many thanks to Jef Wijsen for helping with the slides!

Previous Lecture

- Defined inconsistent databases and repairs
- Defined Consistent Query Answering (CQA)
- Saw a schema with primary-key constraints and a CQ where:
  - CQA can be translated into a formula in First Order Logic (FO) over the inconsistent instance
  - Hence, computable in polynomial time
  - CQA is coNP-hard
  - CQA cannot be phrased in FO over the inconsistent instance, but is still computable in polynomial time

This Lecture

- We will focus on schemas with primary-key constraints, and CQs without self joins
  - That is, CQs where each relation occurs at most once
- We will learn a recent result that shows how to distinguish between the three cases
- Such a result is called a trichotomy, since it classifies all cases into three pairwise-disjoint categories
- We will see how to rewrite CQA into SQL in the case of FO rewritability
In this lecture, we consider only schemas \( S = (R, \Sigma) \) such that \( \Sigma \) consists of primary keys.

That is:

- For every relation name \( R \in R \) there is a unique key constraint \( R : X \rightarrow Y \) in \( \Sigma \)
- There are no other constraints in \( \Sigma \)

Note: “no key” is the same as “left-hand side contains all attributes”.

In our examples we underline the key attributes.

For instance, if \( R \) contains \( R(A, B, C, D) \) then \( R(x, y, z, w) \) means that \( \Sigma \) contains the key constraint \( R : AB \rightarrow CD \).

Recalling Data Complexity

Recall: in data complexity we fix the schema and query, and only the instance \( I \) is considered input.

Effectively, every schema \( S \) and query \( Q \) define a separate computational problem \( P_{S, Q} \).

Consistent Answers

**Definition (Consistent Answers)**

Let \( S = (R, \Sigma) \) be a schema, \( Q \) a query over \( S \), and \( I \) an inconsistent instance over \( S \). A tuple \( a \) is a consistent answer if \( a \in Q(J) \) for every repair \( J \). We denote by \( \text{Consistent}^Q_{\Sigma}(I) \) the set of all consistent answers. Hence, we have:

\[
\text{Consistent}^Q_{\Sigma}(I) = \bigcap_{J \in \text{Repairs}_{\Sigma}(I)} Q(J)
\]

Trichotomy Theorem

**Theorem [KW15]**

Let \( S = (R, \Sigma) \) be a schema, such that \( \Sigma \) consists of primary keys.

Let \( Q \) be a CQ without self joins. Assume that \( \text{P} \neq \text{NP} \). Then exactly one of the following is true.

- \( \text{Consistent}^Q_{\Sigma}(I) \) can be formulated as a query in \( \text{FO} \) (hence, computable in polynomial time).
- \( \text{Consistent}^Q_{\Sigma}(I) \) cannot be formulated as a query in \( \text{FO} \), but is still computable in polynomial time.
- Testing whether \( \text{Consistent}^Q_{\Sigma}(I) \) is empty is \( \text{NP}-\text{complete} \).

Moreover, we can compute in polynomial time (in \( S \) and \( Q \)) in which case we are.
Historical Notes I

- 2005: Fuxman and Miller [FM05] claim a dichotomy for a class of conjunctive queries without self joins
  - A flaw in their proof and result discovered by Wijsen [Wij10b]
- 2010: Wijsen [Wij10a] establishes a dichotomy in FO rewritability for acyclic CQs without self joins
- 2012: Kolaitis and Pema [KP12] prove a dichotomy (P vs coNP-complete) for CQs with two atoms and no self joins

Historical Notes II

- 2013: Fontaine [Fon13] establishes an explanation on why it is difficult to establish dichotomies for (U)CQs with self joins
  - Basically, it entails solving a long standing open problem
- 2014: Koutris and Suciu [KS14] prove a dichotomy for CQs without self joins, where every relation is binary (with a key)
- 2015: Koutris and Wijsen [KW15] prove a trichotomy for all CQs without self joins
  - That is, the trichotomy we learn here

Setup

- Throughout this section, we fix a schema $S = (R, \Sigma)$ and a CQ $Q$
  - $\Sigma$ consists of primary keys (one for each relation)
  - $Q$ has no self joins
  - Recall that for such $\Sigma$, a repair is a maximal consistent subset of $I$
  - We first assume that $Q$ is Boolean (that is, there are no variables in the head)
  - Hence, the goal is to determine whether $Q$ is true in every repair

Example

Let $Q() := R(x, a, y), S(z, w), T(z, y, u), U(b, z)$

- $\text{Atoms}(Q) = \{ R(x, a, y), S(z, w), T(z, y, u), U(b, z) \}$
- $\text{Var}(Q) = \{ x, y, z, w, u \}$
- $\alpha_S = S(z, w)$
- $\alpha = \alpha_S = R(x, a, y) \Rightarrow \alpha_{\text{Var}} = \{ x, y \}$, $\text{KVar}(\alpha) = \{ x \}$
- $\alpha = S(z, w) \Rightarrow \alpha_{\text{Var}} = \{ x, z, w \}$, $\text{KVar}(\alpha) = \{ x \}$

We denote constants by non-italic letters from the beginning of the alphabet (e.g., $a$ and $b$), as opposed to variables (e.g., $x$ and $y$)

Notation

- We denote by:
  - $\text{Atoms}(Q)$ the set of atoms of $Q$
  - $\text{Var}(Q)$ the set of all the variables of $Q$
  - $\alpha_R$ the atom of $Q$ over the relation name $R$
  - $R_{\alpha}$ the relation name of the atom $\alpha$

- For $\alpha \in \text{Atoms}(Q)$, we denote by:
  - $\text{Var}(\alpha)$ the variables that occur in $\alpha$
  - $\text{KVar}(\alpha)$ the variables that occur in key attributes of $R_{\alpha}$
We define the following set of functional dependencies (FDs):

$$\text{FD}(Q) \overset{\text{def}}{=} \{ \text{KVar}(\alpha) \rightarrow \text{Var}(\alpha) \mid \alpha \in \text{Atoms}(Q) \}$$

FD’(Q) denotes the set of all FDs over Var(Q) that are logically implied from FD(Q).

Equivalently (by Armstrong’s axioms), FD’(Q) is obtained from FD(Q) by repeatedly applying the following rules:

- If $x \rightarrow X'$ whenever $X' \subseteq X$ (reflexivity)
- If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$ (transitivity)
- If $X \rightarrow Y$, then $X \cup Z \rightarrow Y \cup Z$ (augmentation)

Hence, in independent of $x$.

If $x$, then $Q() := R(x,y), S(x,z), T(z,y), U(b,z)$.  

Example: $Q() := R(x,a,y), S(x,z), T(z,y), U(b,z)$.  

Observe that every $x \in \text{KVar}(\alpha)$ is an external dependent of $\alpha$.

Example: $Q() := R(x,a,y), S(x,z), T(z,y), U(b,z)$.  

Which variables are external dependents of $\alpha_R$?  

x, z, w  

Which variables are external dependents of $\alpha_S$?  

x, y, z, u

If $x$ is not an external dependent of $\alpha$, then we say that $x$ is externally independent of $\alpha$.

Let $\alpha$ and $\gamma$ be two distinct atoms of $Q$.

We say that $\alpha$ attacks $\gamma$ if there is a sequence $\beta_1, \ldots, \beta_n$ of atoms such that:

- $\alpha = \beta_1$
- $\beta_{n-1} \gamma \beta_n$

Every Var($\beta_i$) \cap Var($\beta_{i+1}$) contains at least one variable that is externally independent of $\alpha$.

If $\beta$ and $\gamma$ are atoms, then we denote by $\beta \overset{\leq}{\sim}_R \gamma$ the fact that $x$ is a variable in Var($\beta$) \cap Var($\gamma$) that is externally independent of $\alpha_R$.

Hence, $\alpha$ attacks $\gamma$ if and only if there exists a sequence $\beta_1 \overset{1}{\sim}_R \beta_2 \overset{2}{\sim}_R \ldots \overset{n}{\sim}_R \beta_n$, where $\beta_1 = \alpha$, $R = R_1$, and $\beta_n = \gamma$.

Examples:

- $Q() := R(x,a,y), S(x,z), T(z,y), U(b,z)$
- $R(x,a,y)$ attacks $T(z,y,u)$:
  $$R(x,a,y) \overset{y}{\sim}_R T(z,y,u)$$
- $U(b,z)$ attacks all other atoms:
  $$U(b,z) \overset{z}{\sim}_U S(x,z), w \overset{z}{\sim}_U R(x,a,y) \overset{y}{\sim}_U T(z,y,u)$$
**Weak and Strong Attack**

- If $\alpha$ attacks $\gamma$ then we say that:
  - $\alpha$ weakly attacks $\gamma$ if $\text{FD}(Q)$ implies $\text{KVar}(\alpha) \rightarrow \text{KVar}(\gamma)$
  - $\alpha$ strongly attacks $\gamma$ otherwise

- Example: $Q() := R(x, a, y), S(x, z, w), T(z, y, u), U(b, z)$
  - $R(x, a, y)$ weakly attacks $T(z, y, u)$
  - $U(b, z)$ strongly attacks all other atoms

**Attack Graphs**

- The attack graph of $Q$ is the directed graph $G = (V, E)$ where:
  - $V$ is Atoms($Q$)
  - There is an edge $(\alpha, \gamma)$ whenever $\alpha$ attacks $\gamma$

- An edge $(\alpha, \gamma)$ is:
  - weak if $\alpha$ weakly attacks $\gamma$ (i.e., $\text{FD}(Q)$ implies $\text{KVar}(\alpha) \rightarrow \text{KVar}(\gamma)$)
  - strong if $\alpha$ strongly attacks $\gamma$

**Example**

$$Q() := R(x, a, y), S(x, z, w), T(z, y, u), U(b, z)$$

**Table of Contents**

- Introduction
- Trichotomy Theorem
- Attacks
- Refined Trichotomy
- FO Rewriting with SQL

**Refined Trichotomy (Boolean case)**

**Theorem [KW15]**

Let $S = (R, \Sigma)$ be a schema, such that $\Sigma$ consists of primary keys. Let $Q$ be a Boolean CQ without self joins, and let $G$ be the attack graph of $Q$.

1. $G$ is acyclic if and only if $\text{Consistent}_Q^S$ is expressible in $\text{FO}$.
2. If $G$ has cycles, but no cycle contains a strong edge, then $\text{Consistent}_Q^S$ can be computed in polynomial time.
3. If $G$ has a cycle with a strong edge, then it is coNP-complete to decide whether $\text{Consistent}_Q^S$ is true on a given instance.

$$Q() := R(x, a, y), S(x, z, w), T(z, y, u), U(b, z)$$

$\Rightarrow$ in $\text{FO}$
**Example 2**

<table>
<thead>
<tr>
<th>LC(lecturer,course)</th>
<th>CT(course,ta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer → course</td>
<td>course → ta</td>
</tr>
</tbody>
</table>

Query: Does any course have both a lecturer and a TA?

$$Q() := \text{LC}(x, y), \text{CT}(y, z)$$

$$\Rightarrow \text{in FO}$$

**Example 3**

<table>
<thead>
<tr>
<th>LC(lecturer,course)</th>
<th>TC(ta,course)</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer → course</td>
<td>ta → course</td>
</tr>
</tbody>
</table>

Query: Does any course have both a lecturer and a TA?

$$Q() := \text{LC}(x, y), \text{TC}(x', y)$$

$$\Rightarrow \text{coNP-complete}$$

**Example 4**

<table>
<thead>
<tr>
<th>LC(lecturer,course)</th>
<th>CT(course,ta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>lecturer → course</td>
<td>course → ta</td>
</tr>
</tbody>
</table>

Query: Does any course have the same lecturer and TA?

$$Q() := \text{LC}(x, y), \text{CT}(y, x)$$

$$\Rightarrow \text{not in FO, but in polynomial time}$$

**Extending to Non-Boolean CQs**

- To extend the trichotomy to non-Boolean CQs, we need some notation.
- If \( x \) is a sequence of variables and \( a \) is a sequence of constants of the same length as \( x \), then \( Q[x \rightarrow a] \) is the CQ that is obtained from \( Q \) by replacing each variable \( x_i \) with \( a_i \)
  - If \( x_i \) is a head variable, then we remove \( x_i \) from the head

**Generalized Trichotomy (non-Boolean case)**

**Theorem [KW15]**

Let \( S = (R, \Sigma) \) be a schema, such that \( \Sigma \) consists of primary keys. Let \( Q(x) \) be a CQ without self joins (where \( x \) is the sequence of head variables). Let \( Q_a \) be the Boolean CQ \( Q[x \rightarrow a] \) for some tuple \( a \) of constants, and let \( G \) be the attack graph of \( Q_a \).

- **G** is acyclic if and only if \( \text{Consistent}_G^{Q_a} \) is equivalent to some \( \varphi(x) \) in FO.
- **If** \( G \) has cycles, but no cycle contains a strong edge, then \( \text{Consistent}_G^{Q_a} \) can be evaluated in polynomial time.
- **If** \( G \) has a cycle with a strong edge, then non-emptiness of \( \text{Consistent}_G^{Q_a} \) is coNP-complete.

**Example**

$$Q(y) := R(x, a, y), S(z, w), T(z, u), U(b, z)$$

$$Q'(y) := R(x, a, c), S(z, w), T(z, c, u), U(b, z)$$

$$\Rightarrow \text{in FO}$$
Notation

- We denote $\alpha \in \text{Atoms}(Q)$ by $\alpha(x, y)$, where $x$ is KVar($\alpha$) and $y$ is Var($\alpha$) \setminus KVar($\alpha$)
- If $\alpha \in \text{Atoms}(Q)$, then $Q^\alpha$ is the CQ obtained from $Q$ by removing $\alpha$
- Recall: if $x$ is a sequence of variables and $\alpha$ is a sequence of constants of the same length as $x$, then $Q_{[x \mapsto \alpha]}$ is the CQ that is obtained from $Q$ by replacing each variable $x_i$ with $\alpha_i$
- It may be the case that $Q$ does not contain some of the $x_i$

Problem Definition

- Let $S = (R, \Sigma)$ be a schema, such that $\Sigma$ consists of primary keys (one per relation name)
- Given: a CQ $Q$ over $R$ such that
  - $Q$ has no self joins (i.e., no relation name occurs more than once)
  - The attack graph of $Q$ is acyclic
- Goal: Compute an SQL query $Q_{\text{cons}}$ over $R$, such that for every inconsistent instance $I$ we have:
  $$\text{Consistent}_{\Sigma}(I) = Q_{\text{cons}}(I)$$

Key Lemma

**Lemma [KW15]**

Let $S = (R, \Sigma)$ be a schema where $\Sigma$ consists of primary keys. Let $Q$ be a Boolean CQ without self joins, and $I$ an inconsistent instance. Let $\alpha(x, y)$ be an atom without incoming edges in the attack graph of $Q$. The following are equivalent.

- $Q$ is consistent (i.e., true on every repair of) over $I$.
- For some $\alpha(a, b) \in I$, the CQ $Q_{[x \mapsto \alpha]}$ is consistent over $I$.
- There is a fact $f = \alpha(a, b)$ in $I$ such that: for all facts $g$ of $R_\alpha$ with the key of $f$ there is $c$ such that: (1) $g = \alpha(a, c)$, and
- $Q_{[x \mapsto (x, y) \mapsto \alpha(a, c)]}$ is consistent over $I$. 
Another Lemma

**Lemma [KW15]**

Let \( S = (R, \Sigma) \) be a schema where \( \Sigma \) consists of primary keys. Let \( Q \) be a Boolean CQ without self joins and with an acyclic attack graph. Let \( \alpha(x, y) \) be an atom without incoming edges in the attack graph of \( Q \). For every \( \alpha(a, c) \in I \), the CQ \( Q' \) of \( Q \) without \( \alpha(a, c) \) has an acyclic attack graph.

Notation

- For a counter \( l \), we denote by:
  - \( R^l \) the sequence obtained from \( R \) by replacing each \( R_i \) with \( "R_i \, R_i^l" \) (i.e., naming \( R_i \) by \( R_i^l \))
  - \( X^l \) the sequence obtained from \( X \) by replacing each \( R_i \) with \( R_i^l \)
  - \( AC^l \) the conjunction obtained from \( AC \) by replacing each \( R_i \) with \( R_i, A = R_i^l, A \)
  - \( TC^l \) the conjunction obtained from \( TC \) by replacing each \( R_i \) with \( R_i, A = t \)

More Notation

- If \( R' \) is a subsequence of \( R \), then we denote by:
  - \( AC \cap R' \) the restriction of \( AC \) to those \( R_i, A = R_i, B \) where \( R_i \in R' \) and \( R_j \in R' \)
  - \( TC \cap R' \) the restriction of \( TC \) to those \( R_i, A = t \) where \( R_i \in R' \)

Selecting a Non-Attacked Atom

- Let \( \alpha \) be a non-attacked atom (i.e., \( \alpha \) has no incoming edges in the attack graph), and let \( R = R_{\alpha} \)
- Denote by:
  - \( K = R.A_1, \ldots, R.A_\ell \) the key attributes of \( R \)
  - \( V = R.B_1, \ldots, R.B_\ell \) the non-key attributes of \( R \)

Recursive Rewriting

- Begin with Boolean: assuming ‘true’ instead of \( X \)
  - \( \text{SELECT} '\text{true}' \text{ FROM} R \text{ WHERE} AC \text{ AND} TC \)
Recursive Rewriting

- Begin with Boolean: assuming 'true' instead of X
  \[
  \text{SELECT 'true' FROM } R \text{ WHERE } AC \text{ AND } TC
  \]
- We create the rewriting $\text{Rewrite}(R, AC, TC)$:
  \[
  \text{SELECT 'true' FROM } R' \text{ WHERE }
  \begin{cases}
    \text{SELECT 'true' FROM } R \cap R' \text{ WHERE } K^2 = K^1 \text{ AND NOT }
    \{\text{AC} \cap \{R^2\} \text{ AND } TC \cap \{R^2\}\} \text{ AND }
    \text{EXISTS}(\text{Rewrite}(R', AC \cap R', TC'))
  \end{cases}
  \]

Example 1

\[
\begin{array}{c}
\text{LC}(Ax, Ay) : \text{CT}(Ay, Az) \\
Q() = \text{LC}(z, y), \text{CT}(y, z)
\end{array}
\]

\[
\begin{array}{c}
\text{SELECT 'true' FROM } \text{LC, CT WHERE } LC.Ay = CT.Ay
\end{array}
\]

Example 2

\[
\begin{array}{c}
\text{LC}(Ax, Ay) : \text{CT}(Ay, Az) \\
Q() = \text{LC}(z, y), \text{CT}(y, Avi)
\end{array}
\]

\[
\begin{array}{c}
\text{SELECT 'true' FROM } \text{LC, CT WHERE } LC.Ay = CT.Ay \text{ AND CT.Az = 'Avi'}
\end{array}
\]
Example 2

SELECT 'true' FROM LC, CT WHERE LC.Ay=CT.Ay AND CT.Az='Avi'

SELECT 'true' FROM LC LC1 WHERE NOT EXISTS (
    SELECT 'true' FROM LC LC2 WHERE LC2.Ax=LC1.Ax AND NOT
    EXISTS(
        SELECT 'true' FROM CT CT1 WHERE NOT EXISTS (
            SELECT 'true' FROM CT CT2 WHERE
            CT2.Ay=CT1.Ay AND NOT
            (CT2.Ay = LC2.Ay AND CT2.Az = 'Avi')
        )
    )
)

48/53

Example 3

SELECT X FROM R WHERE AC AND TC

SELECT X0 FROM R0 WHERE EXISTS (Rewrite(R, AC, TC'))

TC' is obtained from TC by adding \( R_i.A = R_{0i}.A \) for every \( R_i.A \) in X.

48/53

References I

- Gaëlle Fontaine, Why is it hard to obtain a dichotomy for consistent query answering?, LICS, 2013, pp. 550–559.
