Conjunctive Queries: Evaluation and Static Analysis

(Chapter 14 and 15 of DBT)

A match of a conjunctive query \( Q(x_1, \ldots, x_k) \) to a database \( D \) is a homomorphism \( h \) from the set of atoms \( \text{body} \) to the set of atoms \( D \)

The answer to \( Q(x_1, \ldots, x_k) \) over \( D \) is the set of \( k \)-tuples

\[
Q(D) := \{(h(x_1), \ldots, h(x_k)) \mid h \text{ is a match of } Q \text{ in } D\}
\]

The answer consists of the witnesses for the distinguished variables of \( Q \)

Pattern Matching Problem

List the airlines that fly directly from London to Glasgow

\[
\begin{align*}
\text{Flight(VIE,LHR,BA),} & \quad \text{Airport(VIE,Vienna),} \\
\text{Flight(LHR,EDI,BA),} & \quad \text{Airport(LHR,London),} \\
\text{Flight(LGW,GLA,U2),} & \quad \text{Airport(LGW,London),} \\
\text{Flight(LCA,VIE,OS),} & \quad \text{Airport(LCA,Larnaca),} \\
\text{Flight(GLA,LGW,U2),} & \quad \text{Airport(GLA,Glasgow),} \\
\text{Flight(EDI,LCA,OS),} & \quad \text{Airport(EDI,Edinburgh),}
\end{align*}
\]

\[Q(z) : \text{Airport}(x,\text{London}), \text{Airport}(y,\text{Glasgow}), \text{Flight}(x,y,z)\]
Query Evaluation

• Understand the complexity of evaluating a conjunctive query over a database
• What to measure? Queries may have a large output, and it would be misleading to count the output as "complexity"
• We therefore consider the following decision problem for $CQ$ Evaluation

<table>
<thead>
<tr>
<th>CQ Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: a database $D$, a CQ $Q(x_1, \ldots, x_k) : \text{-body}$, and a tuple $(a_1, \ldots, a_k)$ of values</td>
</tr>
<tr>
<td>Question: $(a_1, \ldots, a_k) \in Q(D)$?</td>
</tr>
</tbody>
</table>

Data Complexity of Query Evaluation

• Measures the complexity in terms of the size of the database - the query is fixed
• Meaningful in practice since the database is usually much bigger than the query
• We consider the following decision problem for a fixed CQ $Q(x_1, \ldots, x_k) : \text{-body}$

<table>
<thead>
<tr>
<th>Q-Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: a database $D$, and a tuple $(a_1, \ldots, a_k)$ of values</td>
</tr>
<tr>
<td>Question: $(a_1, \ldots, a_k) \in Q(D)$?</td>
</tr>
</tbody>
</table>

Complexity of Query Evaluation

**Theorem:** $CQ$ Evaluation is NP-complete, and in PTIME in data complexity

**Proof:**

**(NP-membership)** Guess-and-check:

- Consider a database $D$, a CQ $Q(x_1, \ldots, x_k) : \text{-body}$, and a tuple $(a_1, \ldots, a_k)$ of values
- Guess a substitution $h : \text{terms(body)} \rightarrow \text{terms(D)}$
- Verify that $h$ is a match of $Q$ in $D$, i.e., $h(body) \subseteq D$ and $(h(x_1), \ldots, h(x_k)) = (a_1, \ldots, a_k)$

**(NP-hardness)** Reduction from 3-colorability

3COL

**Input:** an undirected graph $G = (V,E)$

**Question:** is there a function $c : V \rightarrow \{R,G,B\}$ such that $(v,u) \in E \Rightarrow c(v) \neq c(u)$?

Therefore, $G$ is 3-colorable iff $G$ can be mapped to $K_3$, i.e., $G \hom K_3$, the Boolean CQ that represents $G$
Complexity of Query Evaluation

**Theorem:** CQ-Evaluation is NP-complete, and in PTIME in data complexity

**Proof:**

**(NP-membership)** Guess-and-check:
- Consider a database $D$, a CQ $Q(x_1, ..., x_k) \rightarrow \text{body}$, and a tuple $(a_1, ..., a_k)$ of values
- Guess a substitution $h : \text{terms(body)} \rightarrow \text{terms(D)}$
- Verify that $h$ is a match of $Q$ in $D$, i.e., $h(\text{body}) \subseteq D$ and $(h(x_1), ..., h(x_k)) = (a_1, ..., a_k)$

**(NP-hardness)** Reduction from 3-colorability

**(in PTIME)** For every substitution $h : \text{terms(body)} \rightarrow \text{terms(D)}$, check if $h(\text{body}) \subseteq D$ and $(h(x_1), ..., h(x_k)) = (a_1, ..., a_k)$

---

**Static Analysis**

**CQ-Satisfiability**

*Input:* a conjunctive query $Q$

*Question:* is there a database $D$ such that $Q(D)$ is non-empty?

- If the answer is no, then the input query $Q$ makes no sense
- CQ-Evaluation becomes trivial - the answer is always NO!

---

**Static Analysis**

**CQ-Equivalence**

*Input:* two conjunctive queries $Q_1$ and $Q_2$

*Question:* $Q_1 \equiv Q_2$? or $Q_1(D) = Q_2(D)$ for every database $D$?

- Replace a query $Q_2$ with a query $Q_1$ that is easier to evaluate
- But, we have to be sure that $Q_1(D) = Q_2(D)$ for every database $D$

---

**Static Analysis**

**CQ-Containment**

*Input:* two conjunctive queries $Q_1$ and $Q_2$

*Question:* $Q_1 \subseteq Q_2$? or $Q_1(D) \subseteq Q_2(D)$ for every database $D$?

- Equivalence boils down to two containment checks
- Clearly, $Q_1(D) = Q_2(D)$ if $Q_1(D) \subseteq Q_2(D)$ and $Q_2(D) \subseteq Q_1(D)$
Complexity of Static Analysis

CQ-Satisfiability
Input: a conjunctive query \( Q \)
Question: is there a database \( D \) such that \( Q(D) \) is non-empty?

CQ-Equivalence
Input: two conjunctive queries \( Q_1 \) and \( Q_2 \)
Question: \( Q_1 \equiv Q_2 \) or \( Q_1(D) = Q_2(D) \) for every database \( D \)?

CQ-Containment
Input: two conjunctive queries \( Q_1 \) and \( Q_2 \)
Question: \( Q_1 \subseteq Q_2 \) or \( Q_1(D) \subseteq Q_2(D) \) for every database \( D \)?

Canonical Database

• Convert a conjunctive query \( Q \) into a database \( D(Q) \) - the canonical database of \( Q \)

• Given a conjunctive query of the form \( Q(x) \) : body, \( D(Q) \) is obtained from body by replacing each variable \( x \) with a new value \( c(x) = x \).

• E.g., given \( Q(x,y) : R(x,y), P(y,z,w), R(z,x) \), then \( D(Q) = \{R(x,y), P(y,z,w), R(z,x)\} \)

• Note: The mapping \( c : \) \{variables in body\} → \{new values\} is a bijection, where \( c(\text{body}) = D(Q) \) and \( c^{-1}(D(Q)) = \text{body} \)

Satisfiability of CQs

CQ-Satisfiability
Input: a conjunctive query \( Q \)
Question: is there a database \( D \) such that \( Q(D) \) is non-empty?

Theorem: A conjunctive query \( Q \) is always satisfiable

Proof: Due to its canonical database - \( Q(D(Q)) \) is trivially non-empty

Equivalence and Containment of CQs

CQ-Equivalence
Input: two conjunctive queries \( Q_1 \) and \( Q_2 \)
Question: \( Q_1 \equiv Q_2 \) or \( Q_1(D) = Q_2(D) \) for every database \( D \)?

CQ-Containment
Input: two conjunctive queries \( Q_1 \) and \( Q_2 \)
Question: \( Q_1 \subseteq Q_2 \) or \( Q_1(D) \subseteq Q_2(D) \) for every database \( D \)?

\[ Q_1 \equiv Q_2 \iff Q_1 \subseteq Q_2 \text{ and } Q_2 \subseteq Q_1 \]

\[ Q_1 \subseteq Q_2 \iff Q_1 \equiv (Q_1 \land Q_2) \]

...thus, we can safely focus on CQ-Containment
**Homomorphism Theorem**

A query homomorphism from $Q_1(x_1,...,x_k) : \text{body}_1$ to $Q_2(y_1,...,y_l) : \text{body}_2$ is a substitution $h : \text{terms(body}_1) \rightarrow \text{terms(body}_2)$ such that:

1. $h$ is a homomorphism from $\text{body}_1$ to $\text{body}_2$.
2. $(h(x_1),...,h(x_k)) = (y_1,...,y_l)$

**Homomorphism Theorem: Proof**

Assume that $Q_1(x_1,...,x_k) : \text{body}_1$ and $Q_2(y_1,...,y_l) : \text{body}_2$.

$(\Rightarrow)$ $Q_1 \subseteq Q_2 \Rightarrow$ there exists a query homomorphism from $Q_1$ to $Q_2$.

- Clearly, $(c(x_1),...,c(x_k)) \in Q_1[D(Q_1)]$ - recall that $Q_1[D(Q_1)] = c(\text{body}_1)$.
- Since $Q_1 \subseteq Q_2$, we conclude that $(c(x_1),...,c(x_k)) \in Q_2[D(Q_2)]$.
- Therefore, there exists a homomorphism $h$ such that $h(\text{body}_1) \subseteq Q_2[D(Q_2)] = c(\text{body}_2)$ and $h(y_1,...,y_l) = (c(x_1),...,c(x_k))$.
- By construction, $c \circ h = \text{body}_1$.
- Therefore, $c \circ h$ is a query homomorphism from $Q_1$ to $Q_2$.

**Homomorphism Theorem: Example**

Consider a database $D$ and a tuple $t$ such that $t \in Q_1[D]$.

- We need to show that $t \in Q_2[D]$.
- Clearly, there exists a homomorphism $g$ such that $g(\text{body}_1) \subseteq D$ and $g(x_1,...,x_k) = t$.
- By hypothesis, there exists a query homomorphism $h$ from $Q_1$ to $Q_2$.
- Therefore, $g \circ h(\text{body}_1) \subseteq D$ and $g(h(x_1,...,x_k)) = t$, which implies that $t \in Q_2[D]$.
Theorem: Let $Q_1$ and $Q_2$ be conjunctive queries. The problem of deciding whether there exists a query homomorphism from $Q_2$ to $Q_1$ is NP-complete.

Proof:

(NP-membership) Guess a substitution, and verify that it is a query homomorphism.

(NP-hardness) Easy reduction from CQ-Evaluation.

By applying the homomorphism theorem we get that:

Corollary: CQ-Equivalence and CQ-Containment are NP-complete.