

## Conjunctive Queries: Evaluation and Static Analysis

(Chapter 14 and 15 of DBT)

[DBT] Database Theory, <https://github.com/pdm-book/community>

## Semantics of Conjunctive Queries

- A **match** of a conjunctive query  $Q(x_1, \dots, x_k) :- \text{body}$  in a database  $D$  is a homomorphism  $h$  from the set of atoms **body** to the set of atoms  $D$
- The **answer** to  $Q(x_1, \dots, x_k) :- \text{body}$  over  $D$  is the set of  $k$ -tuples  

$$Q(D) := \{h(x_1), \dots, h(x_k) \mid h \text{ is a match of } Q \text{ in } D\}$$
- 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

Flight(VIE, LHR, BA),	Airport(VIE, Vienna),
Flight(LHR, EDI, BA),	Airport(LHR, London),
Flight(LGW, GLA, U2),	Airport(LGW, London),
Flight(LCA, VIE, OS),	Airport(LCA, Larnaca),
	Airport(GLA, Glasgow),
	Airport(EDI, Edinburgh)

$Q(z) :- \text{Airport}(x, \text{London}), \text{Airport}(y, \text{Glasgow}), \text{Flight}(x, y, z)$

## Pattern Matching Problem

List the airlines that fly directly from London to Glasgow

Flight(VIE, LHR, BA),	Airport(VIE, Vienna),
Flight(LHR, EDI, BA),	Airport(LHR, London),
<b>Flight(LGW, GLA, U2),</b>	<b>Airport(LGW, London),</b>
Flight(LCA, VIE, OS),	Airport(LCA, Larnaca),
	<b>Airport(GLA, Glasgow),</b>
	Airport(EDI, Edinburgh)

$\{x \mapsto \text{LGW}, y \mapsto \text{GLA}, z \mapsto \text{U2},$   
 $\text{London} \mapsto \text{London}, \text{Glasgow} \mapsto \text{Glasgow}\}$

$Q(z) :- \text{Airport}(x, \text{London}), \text{Airport}(y, \text{Glasgow}), \text{Flight}(x, y, z)$

## 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**

**CQ-Evaluation**

**Input:** a database  $D$ , a CQ  $Q(x_1, \dots, x_k) :- \text{body}$ , and a tuple  $(a_1, \dots, a_k)$  of values

**Question:**  $(a_1, \dots, a_k) \in Q(D)$ ?

*combined complexity*

## 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  $Q(x_1, \dots, x_k) :- \text{body}$

**Q-Evaluation**

**Input:** a database  $D$ , and a tuple  $(a_1, \dots, a_k)$  of values

**Question:**  $(a_1, \dots, a_k) \in Q(D)$ ?

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

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


## NP-hardness

**(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  $(v, u) \in E \Rightarrow c(v) \neq c(u)$ ?

**Lemma:**  $G$  is 3-colorable iff  $G$  can be mapped to  $K_3$ , i.e.,  $G \xrightarrow{\text{hom}}$  

therefore,  $G$  is 3-colorable iff there is a match of  $Q_G$  in  $D = \{E(x, y), E(y, z), E(z, x)\}$

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

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

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

## 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  $Q_1$  with a query  $Q_2$  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,  $Q_1(D) = Q_2(D)$  iff  $Q_1(D) \subseteq Q_2(D)$  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  $Q(\mathbf{x}) :- \text{body}$ ,  $D[Q]$  is obtained from **body** by replacing each variable  $x$  with a new value  $c(x) = \underline{x}$
- E.g., given  $Q(x,y) :- R(x,y), P(y,z,w), R(z,x)$ , then  $D[Q] = \{R(\underline{x}, \underline{y}), P(\underline{y}, \underline{z}, \underline{w}), R(\underline{z}, \underline{x})\}$
- Note:** The mapping  $c : \{\text{variables in body}\} \rightarrow \{\text{new values}\}$  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  $D$ ?

$$Q_1 \equiv Q_2 \text{ iff } Q_1 \subseteq Q_2 \text{ and } Q_2 \subseteq Q_1$$

$$Q_1 \subseteq Q_2 \text{ iff } Q_1 \equiv (Q_1 \wedge Q_2)$$

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

## Homomorphism Theorem

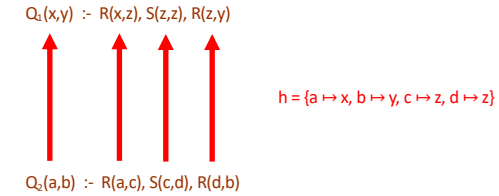
A **query homomorphism** from  $Q_1(x_1, \dots, x_k) \text{ :- } \text{body}_1$  to  $Q_2(y_1, \dots, y_k) \text{ :- } \text{body}_2$  is a substitution  $h : \text{terms}(\text{body}_1) \rightarrow \text{terms}(\text{body}_2)$  such that:

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

**Homomorphism Theorem:** Let  $Q_1$  and  $Q_2$  be conjunctive queries. It holds that:

$Q_1 \subseteq Q_2$  iff there exists a query homomorphism from  $Q_1$  to  $Q_2$

## Homomorphism Theorem: Example



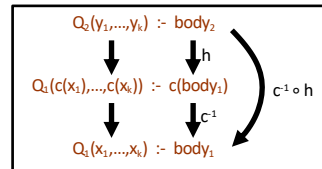
- $h$  is a query homomorphism from  $Q_2$  to  $Q_1 \Rightarrow Q_1 \subseteq Q_2$
- But, there is no homomorphism from  $Q_1$  to  $Q_2 \Rightarrow Q_1 \subsetneq Q_2$

## Homomorphism Theorem: Proof

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

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

- Clearly,  $(c(x_1), \dots, c(x_k)) \in Q_1(D[Q_1])$  - recall that  $D[Q_1] = c(\text{body}_1)$
- Since  $Q_1 \subseteq Q_2$ , we conclude that  $(c(x_1), \dots, c(x_k)) \in Q_2(D[Q_1])$
- Therefore, there exists a homomorphism  $h$  such that  $h(\text{body}_2) \subseteq D[Q_1] = c(\text{body}_1)$  and  $h((y_1, \dots, y_k)) = (c(x_1), \dots, c(x_k))$
- By construction,  $c^{-1}(c(\text{body}_1)) = \text{body}_1$  and  $c^{-1}((c(x_1), \dots, c(x_k))) = (x_1, \dots, x_k)$
- Therefore,  $c^{-1} \circ h$  is a query homomorphism from  $Q_2$  to  $Q_1$

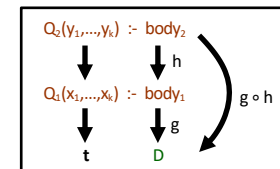


## Homomorphism Theorem: Proof

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

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

- 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  $g((x_1, \dots, x_k)) = t$
- By hypothesis, there exists a query homomorphism  $h$  from  $Q_2$  to  $Q_1$
- Therefore,  $g(h(\text{body}_2)) \subseteq D$  and  $g(h((y_1, \dots, y_k))) = t$ , which implies that  $t \in Q_2(D)$



## Existence of a Query Homomorphism

**Theorem:** Let  $Q_i$  and  $Q_j$  be conjunctive queries. The problem of deciding whether there exists a query homomorphism from  $Q_j$  to  $Q_i$  is NP-complete

**Proof:**

**(NP-membership)** Guess a substitution, and verify that 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