Schema Mappings and Data Exchange

Lecture #3

EASSLC 2012

Southwest University

August 2012

Query Languages for the Relational Data Model

Codd introduced two different query languages for the relational data model:

- Relational Algebra, which is a procedural language.
  - It is an algebraic formalism in which queries are expressed by applying a sequence of operations to relations.
- Relational Calculus, which is a declarative language.
  - It is a logical formalism in which queries are expressed as formulas of first-order logic.

Codd's Theorem: Relational Algebra and Relational Calculus are "essentially equivalent" in terms of expressive power.

(but what does this really mean?)

Relational Algebra

- Definition: A relational algebra expression is a string obtained from relation schemas using union, difference, cartesian product, projection, and selection.

- Context-free grammar for relational algebra expressions:

  $E := R, S, \ldots \mid (E_1 \cup E_2) \mid (E_1 - E_2) \mid (E_1 \times E_2) \mid \pi_L(E) \mid \sigma_\Theta(E)$,

  where
  - R, S, … are relation schemas
  - L is a list of attributes
  - Θ is a condition.

Relational Calculus (First-Order Logic for Databases)

- First-order variables: $x, y, z, \ldots, x_1, \ldots, x_k, \ldots$
  - They range over values that may occur in tables.
- Relation symbols: R, S, T, … of specified arities (names of relations)
- Atomic (Basic) Formulas:
  - $R(x_1,\ldots,x_k)$, where R is a k-ary relation symbol (alternatively, $(x_1,\ldots,x_k) \in R$; the variables need not be distinct)
  - (x op y), where op is one of $=, \neq, <, >, \leq, \geq$
  - (x op c), where c is a constant and op is one of $=, \neq, <, >, \leq, \geq$.
- Relational Calculus Formulas:
  - Every atomic formula is a relational calculus formula.
  - If φ and ψ are relational calculus formulas, then so are:
    - $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $\neg \psi$, $(\varphi \rightarrow \psi)$ (propositional connectives)
    - $(\exists x\, \varphi)$ (existential quantification)
    - $(\forall x\, \varphi)$ (universal quantification).

Relational Calculus as a Database Query Language

Definition:

- A relational calculus expression is an expression of the form

  $$\{ (x_1,\ldots,x_k):\ \varphi(x_1,\ldots x_k) \},$$

  where $\varphi(x_1,\ldots,x_k)$ is a relational calculus formula with $x_1,\ldots,x_k$ as its free variables.
- When applied to a relational database I, this relational calculus expression returns the k-ary relation that consists of all k-tuples $(a_1,\ldots,a_k)$ that make the formula "true" on I.

Example: The relational calculus expression

$$\{ (x,y):\ \exists z(E(x,z) \wedge E(z,y)) \}$$

returns the set P of all pairs of nodes (a,b) that are connected via a path of length 2.

Equivalence of Relational Algebra and Relational Calculus

Theorem: The following are equivalent for a k-ary query q:

1. There is a relational algebra expression E such that q(I) = E(I), for every database instance I
   (in other words, q is expressible in relational algebra).

2. There is a relational calculus formula ψ such that q(I) = ψ^{adom} (I)
   (in other words, q is expressible in relational calculus under the active domain interpretation).

Queries

Definition: Let **S** be a relational database schema.

- A k-ary query on **S** is a function q defined on database instances over S such that if I is a database instance over S, then q(I) is a k-ary relation on adom(I) that is invariant under isomorphisms (i.e., if h: I $\rightarrow$ J is an isomorphism, then q(J) = h(q(I)).

- A Boolean query on **S** is a function q defined on database instances over S such that if I is a database instance over S, then q(I) = 0 or q(I) = 1, and q(I) is invariant under isomorphisms.

Example: The following are Boolean queries on graphs:

- Given a graph E (binary relation), is the diameter of E at most 3?
- Given a graph E (binary relation), is E connected?

Three Fundamental Algorithmic Problems about Queries

- The Query Evaluation Problem: Given a query q and a database instance I, find q(I).

- The Query Equivalence Problem: Given two queries q and q′ of the same arity, is it the case that $q \equiv q'$?

  (i.e., is it the case that, for every database instance I, we have that q(I) = q′(I)?)

- The Query Containment Problem: Given two queries q and q′ of the same arity, is it the case that $q \subseteq q'$?

  (i.e., is it the case that, for every database instance I, we have that $q(I) \subseteq q'(I)$?)

Summary

- Relational Algebra and Relational Calculus have “essentially” the same expressive power.

- The Query Equivalence Problem for Relational Calculus in undecidable.

- The Query Containment Problem for Relational Calculus is undecidable.

- The Query Evaluation Problem for Relational Calculus is PSPACE-complete.

Sublanguages of Relational Calculus

- Question: Are there interesting sublanguages of relational calculus for which the Query Containment Problem and the Query Evaluation Problem are "easier" than the full relational calculus?

- Answer:
  - Yes, the language of conjunctive queries is such a sublanguage.
  - Moreover, conjunctive queries are the most frequently asked queries against relational databases.

Conjunctive Queries

- **Definition:** A conjunctive query is a query expressible by a relational calculus formula in prenex normal form built from atomic formulas $R(y_1,\ldots,y_n)$, and $\wedge$ and $\exists$ only.

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

  where $\chi(x_1, \ldots,x_k, z_1,\ldots,z_k)$ is a conjunction of atomic formulas of the form $R(y_1,\ldots,y_m)$.
  - Equivalently, a conjunctive query is a query expressible by a relational algebra expression of the form

    $\pi_X(\sigma_\Theta(R_1 \times \ldots \times R_n))$, where

    Θ is a conjunction of equality atomic formulas (equijoin).
  - Equivalently, a conjunctive query is a query expressible by an SQL expression of the form

    SELECT <list of attributes>

    FROM <list of relation names>

    WHERE <conjunction of equalities>

Conjunctive Queries

- Definition: A conjunctive query is a query expressible by a relational calculus formula in prenex normal form built from atomic formulas $R(y_1,\ldots,y_n)$, and $\wedge$ and $\exists$ only.

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

  - A conjunctive query can be written as a logic-programming rule:

    $$Q(x_1,\ldots,x_k) \text{ :-- } R_1(\mathbf{u}_1), \ldots, R_n(\mathbf{u}_n), \text{ where}$$

    - Each variable x_i occurs in the right-hand side of the rule.
    - Each $\mathbf{u}_i$ is a tuple of variables (not necessarily distinct)
    - The variables occurring in the right-hand side (the body), but not in the left-hand side (the head) of the rule are existentially quantified (but the quantifiers are not displayed).
    - "," stands for conjunction.

Conjunctive Queries

Examples:

- Path of Length 2: (Binary query)

  $\{(x,y) : \exists z\,(E(x,z) \wedge E(z,y))\}$

  - As a relational algebra expression,

    $\pi_{1,4}(\sigma_{\$2 = \$3}(E \times E))$

  - As a rule:

    q(x,y) :-- E(x,z), E(z,y)

- Cycle of Length 3: (Boolean query)

  $\exists x \exists y \exists z (E(x,y) \wedge E(y,z) \wedge E(z,x))$

  - As a rule (the head has no variables)
    - Q :-- E(x,z), E(z,y), E(z,x)

Conjunctive Queries

- Every relational join is a conjunctive query:
  P(A,B,C), R(B,C,D) two relation symbols
  - $P \bowtie R = \{(x,y,z,w):\ P(x,y,z) \wedge R(y,z,w)\}$
  - q(x,y,z,w) :-- P(x,y,z), R(y,z,w)
    (no variables are existentially quantified)
  - SELECT P.A, P.B, P.C, R.D
    FROM P, R
    WHERE P.B = R.B AND P.C = R.C
- Conjunctive queries are also known as SPJ-queries
  (SELECT-PROJECT-JOIN queries)

Conjunctive Query Evaluation and Containment

- Definition: Two fundamental problems about CQs
  - Conjunctive Query Evaluation (CQE):
    Given a conjunctive query q and an instance I, find q(I).

  - Conjunctive Query Containment (CQC):
    - Given two k-ary conjunctive queries q_1 and q_2,
      is it true that $q_1 \subseteq q_2$?
      (i.e., for every instance I, we have that $q_1(I) \subseteq q_2(I)$)
    - Given two Boolean conjunctive queries q_1 and q_2, is it true that $q_1 \models q_2$? (that is, for all I, if $I \models q_1$, then $I \models q_2$)?
      CQC is logical implication.

CQE vs. CQC

- Recall that for relational calculus queries:
  - The Query Evaluation Problem is decidable (in fact, it is PSPACE-complete).
  - The Query Containment Problem is undecidable.
- Theorem**:** Chandra & Merlin, 1977
  - CQE and CQC are the "same" problem.
  - Moreover, both are decidable (in fact, they are NP-complete).
- Question: What is the common link?
- Answer: The Homomorphism Problem

Isomorphisms Between Database Instances

- Definition: Let I and J be two database instances over the same relational schema **S**.
  - An isomorphism h: I $\rightarrow$ J is a function h: adom(I) $\rightarrow$ adom(J) such that
    - h is one-to-one and onto.
    - For every relational symbol P of S and every $(a_1,\ldots,a_m)$, we have that

      $(a_1,\ldots,a_m) \in P^I$ if and only if $(h(a_1), .., h(a_m)) \in P^J$.
  - I and J are isomorphic if an isomorphism h from I to J exists.
- Note: Intuitively, two database instances are isomorphic if one can be obtained from the other by renaming the elements of its active domain in a 1-1 way.

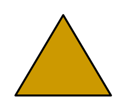
Homomorphisms

- Definition: Let I and J be two database instances over the same relational schema S.
  A homomorphism h: I $\rightarrow$ J is a function h: adom(I) $\rightarrow$ adom(J) such That for every relational symbol P of S and every $(a_1,\ldots,a_m)$, we have that

  $$\text{if } (a_1,\ldots,a_m) \in P^I \text{, then } (h(a_1), .., h(a_m)) \in P^J.$$

- Note: The concept of homomorphism is a relaxation of the concept of isomorphism, since every isomorphism is also a homomorphism, but not vice versa.

- Example:
  - A graph G = (V,E) is 3-colorable
    if and only if
    there is a homomorphism h: G $\rightarrow$ K_3



Homomorphisms

- Fact: Homomorphisms compose, i.e.,
  if f: I $\rightarrow$ J and g: J $\rightarrow$ K are homomorphisms, then
  g$\circ$f: I $\rightarrow$ K is a homomorphims, where g$\circ$f(a) = g(f(a)).

- Definition:
  - Two database instances I and I' are homomorphically equivalent if there is a homomorphism h: I $\rightarrow$ I' and a homomorphism h': I' $\rightarrow$ I.
  - I $\equiv_h$ I' means that I and I' are homomorphically equivalent.

- Note: I $\equiv_h$ I' does **not** imply that I and I' are isomorphic.

Homomorphisms

- Fact: Homomorphisms compose, i.e.,
  if f: I $\rightarrow$ J and g: J $\rightarrow$ K are homomorphisms, then
  g$\circ$f: I $\rightarrow$ K is a homomorphims, where g$\circ$f(a) = g(f(a)).

- Definition:
  - Two database instances I and I′ are homomorphically equivalent if there is a homomorphism h: I $\rightarrow$ I′ and a homomorphism h′: I′ $\rightarrow$ I.
  - I $\equiv_h$ I′ means that I and I′ are homomorphically equivalent.

- Note: I $\equiv_h$ I′ does **not** imply that I and I′ are isomorphic.

The Homomorphism Problem

- Definition: The Homomorphism Problem
  Given two database instances I and J, is there a homomorphism $h: I \rightarrow J$?

- Notation: $I \rightarrow J$ denotes that a homomorphism from I to J exists.

- Theorem: The Homomorphism Problem is NP-complete
  Proof: Easy reduction from 3-Colorabilty
  G is 3-colorable if and only if $G \rightarrow K_3$.

- Exercise: Formulate 3SAT as a special case of the Homomorphism Problem.

The Homomorphism Problem

- Note: The Homomorphism Problem is a fundamental algorithmic problem:
  - Satisfiability can be viewed as a special case of it.
  - k-Colorability can be viewed as a special case of it.
  - Many AI problems, such as planning, can be viewed as a special case of it.
  - In fact, every constraint satisfaction problem can be viewed as a special case of the Homomorphism Problem
    (Feder and Vardi – 1993).

The Homomorphism Problem and Conjunctive Queries

- Theorem: Chandra & Merlin, 1977
  - CQE and CQC are the "same" problem.

- Question: What is the common link?
- Answer:
  - Both CQE and CQC are "equivalent" to the Homomorphism Problem.
  - The link is established by bringing into the picture
    - Canonical conjunctive queries and
    - Canonical database instances.

Canonical CQs and Canonical Instances

- Definition: Canonical Conjunctive Query
  Given an instance $I = (R_1, \ldots, R_m)$, the canonical CQ of I is the Boolean conjunctive query Q^I with (a renaming of) the elements of I as variables and the facts of I as conjuncts, where a fact of I is an expression
  $R_i(a_1,\ldots,a_m)$ such that $(a_1,\ldots,a_m) \in R_i$.

- Example:
  I consists of E(a,b), E(b,c), E(c,a)

  - Q^I is given by the rule:
    Q^I :-- E(x,z), E(z,y), E(y,x)
  - Alternatively, Q^I is
    $$\exists x \exists y \exists z (E(x,z) \wedge E(z,y) \wedge E(y,x))$$

Canonical Conjunctive Query

- Example: K_3, the complete graph with 3 nodes

  K_3 is a database instance with one binary relation E, where

  E = {(b,r), (r,b), (b,g), (g,b), (r,g), (g,r)}

- The canonical conjunctive query Q^{K_3} of K_3 is given by the rule:

  Q^{K_3} :- E(x,y),E(y,x),E(x,z),E(z,x),E(y,z),E(z,y)

- The canonical conjunctive query Q^{K_3} of K_3 is also given by the relational calculus expression:

  $\exists x,y,z(E(x,y) \wedge E(y,x) \wedge E(x,z) \wedge E(z,x) \wedge E(y,z) \wedge E(z,y))$

Canonical Database Instance

- Definition: Canonical Instance

  Given a CQ Q, the canonical instance of Q is the instance I^Q with the variables of Q as elements and the conjuncts of Q as facts.

- Example:

  Conjunctive query Q :-- E(x,y),E(y,z),E(z,w)

  - Canonical instance I^Q consists of the facts E(x,y), E(y,z),E(z,w).
  - In other words, E^{I^Q} = {(x,y), (y,z), (z,w)}.

Canonical Database Instance

- Example:

  Conjunctive query Q(x,y) :-- E(x,z),E(z,y),P(z)

  or, equivalently,

  $$\{(x,y):\ \exists\, z(E(x,z) \wedge E(z,y) \wedge P(z)\}$$

  - Canonical instance I^Q consists of the facts E(x,z), E(z,y),P(z).

  - In other words, $E^{I^Q} = \{(x,z), (z,y)\}$ and $P^{I^Q}=\{z\}$

Canonical Conjunctive Queries and Canonical Instances

- Fact:
  - For every database instance I, we have that $I \models Q^I$.
  - For every Boolean conjunctive query Q, we have that $I^Q \models Q$.

- Fact: Let I be a database instance, let Q^I be its canonical conjunctive query and let I^{Q^I} be the canonical instance of Q^I. Then I is isomorphic to I^{Q^I}.

Canonical Conjunctive Queries and Canonical Instances

Magic Lemma: Assume that Q is a Boolean conjunctive query and J is a database instance. Then the following statements are equivalent.

- $J \models Q$.
- There is a homomorphism $h: I^Q \rightarrow J$.

Proof: Let Q be $\exists x_1 \ldots \exists x_m \, \varphi(x_1,\ldots,x_m)$.

1. $\Rightarrow$ 2. Assume that $J \models Q$. Hence, there are elements $a_1, \ldots, a_m$ in adom(J) such that $J \models \varphi(a_1,\ldots,a_m)$. The function h with $h(x_i) = a_i$, for $i=1,\ldots,m$, is a homomorphism from I^Q to J.

2. $\Rightarrow$ 1. Assume that there is a homomorphism $h: I^Q \rightarrow J$.
Then the values $h(x_i) = a_i$, for $i = 1,\ldots, m$, give values for the interpretation of the existential quantifiers $\exists x_i$ of Q in adom(J) so that $J \models \varphi(a_1,\ldots,a_m)$.

Homomorphisms, CQE, and CQC

The Homomorphism Theorem: Chandra & Merlin – 1977
For Boolean CQs Q and Q′, the following are equivalent:

- $Q \subseteq Q'$
- There is a homomorphism $h: I^{Q'} \rightarrow I^{Q}$
- $I^{Q} \models Q'$.

In dual form:

The Homomorphism Theorem: Chandra & Merlin – 1977
For instances I and I′, the following are equivalent:

- There is a homomorphism $h: I \rightarrow I'$
- $I' \models Q^{I}$
- $Q^{I'} \subseteq Q^{I}$

Homomorphisms, CQE, and CQC

The Homomorphism Theorem: Chandra & Merlin – 1977

For Boolean CQs Q and Q′, the following are equivalent:

1. $Q \subseteq Q'$
2. There is a homomorphism h: $I^{Q'} \rightarrow I^{Q}$
3. $I^{Q} \models Q'$.

Proof:

1. ⇒ 2. Assume $Q \subseteq Q'$. Since $I^{Q} \models Q$, we have that $I^{Q} \models Q'$.

Hence, by the Magic Lemma, there is a homomorphism from $I^{Q'}$ to I^{Q}.

2. ⇒ 3. It follows from the other direction of the Magic Lemma.

3. ⇒ 1. Assume that $I^{Q} \models Q'$. So, by the Magic Lemma, there is a homomorphism h: $I^{Q'} \rightarrow I^{Q}$. We have to show that if $J \models Q$, then $J \models Q'$. Well, if $J \models Q$, then (by the Magic Lemma), there is a homomorphism h′: $I^{Q} \rightarrow J$. The composition $h' \circ h$: $I^{Q'} \rightarrow J$ is a homomorphism, hence (once again by the Magic Lemma!), we have that $J \models Q'$.

Illustrating the Homomorphism Theorem

- **Example:**
  - Q: $\exists x_1 \exists x_2 \exists x_3 \exists x_4 \, (E(x_1,x_2) \wedge E(x_2,x_3) \wedge E(x_3,x_4))$
  - Q′ : $\exists x_1 \exists x_2 \exists x_3 \, (E(x_1,x_2) \wedge E(x_2,x_3))$

  Then:
- $Q \subseteq Q'$

  Homomorphism h: $I^{Q'} \rightarrow I^{Q}$ with

  $h(x_1) = x_1$, $h(x_2) = x_2$, $h(x_3) = x_3$.

- $Q' \not\subseteq Q$ (why?).

Illustrating the Homomorphism Theorem

- **Example:**
  - Q : $\exists x_1 \exists x_2\, (E(x_1,x_2) \wedge E(x_2,x_1))$
  - Q′: $\exists x_1 \exists x_2 \exists x_3 \exists x_4\, (E(x_1,x_2) \wedge E(x_2,x_1) \wedge E(x_2,x_3) \wedge E(x_3,x_2) \wedge E(x_3,x_4) \wedge E(x_4,x_3) \wedge E(x_4,x_1) \wedge E(x_1,x_4))$

  Then:
- $Q \subseteq Q'$

  Homomorphism h: $I^{Q'} \rightarrow I^{Q}$ with

  $h(x_1) = x_1$, $h(x_2) = x_2$, $h(x_3) = x_1$, $h(x_4) = x_2$.
- $Q' \subseteq Q$

  Homomorphism h′: $I^{Q} \rightarrow I^{Q'}$ with $h'(x_1) = x_1$, $h(x_2) = x_2$.
- Hence, $Q \equiv Q'$.

Illustrating the Homomorphism Theorem

Example: 3-Colorability

For a graph G=(V,E), the following are equivalent:

- G is 3-colorable
- There is a homomorphism h: $G \rightarrow K_3$
- $K_3 \models Q^G$
- $Q^{K_3} \subseteq Q^G$.