Database Theory: Conjunctive Queries & Static Analysis

CS 645
Feb 9, 2010
Life of a database theoretician

• Expressiveness of query languages
  – Any query in L1 can be expressed in L2
  – Query q cannot be expressed in L

• Complexity of languages
  – Bounds on resources required to evaluate any query in language L

• Static analysis of queries (for optimization)
  – Given q in L: is it minimal?
  – Given q1 and q2 in L: are they equivalent?

• Views
Coming lectures

• TODAY:
  – Overview of languages
  – Conjunctive queries (CQs)
  – Properties of CQs
  – Containment/equivalence for CQs

• Next lecture
  – Adding recursion
  – Reasoning about views
Query languages

• So far we’ve seen:
  – Relational algebra
  – Relational calculus
  – SQL
Review: relational algebra

• Five operators:
  – Union: $\bigcup$
  – Difference: $-$
  – Selection: $\sigma$
  – Projection: $\Pi$
  – Cartesian Product: $\times$

• Derived or auxiliary operators:
  – Intersection, complement
  – Joins (natural, equi-join, theta join)
  – Renaming: $\rho$
Review: relational calculus

English: Name and sid of students who are taking the course “DB”

RA: \[ \Pi_{\text{name}, \text{sid}} (\text{Students} \bowtie \text{Takes} \bowtie \sigma_{\text{name} = "DB"} (\text{Course})) \]

RC: \{ x_{\text{name}}, x_{\text{sid}} | \exists x_{\text{cid}} \exists x_{\text{term}} \ \text{Students}(x_{\text{sid}}, x_{\text{name}}) \land \text{Takes}(x_{\text{sid}}, x_{\text{cid}}) \land \text{Course}(x_{\text{cid}}, "DB", x_{\text{term}}) \} \]
Review: SQL

Basic form:

```
SELECT attributes
FROM relations (possibly multiple, joined)
WHERE conditions (selections)
```
Query language classes

Recursive Queries

FO queries

Conjunctive Queries

Expressiveness

Algebra
Logic
SQL

RA
(safe) RC
SFW +
UNION
EXCEPT

single datalog rule
Conjunctive Queries

*abbreviated: CQ*

- A **subset** of FO queries (i.e. less expressive)
- Many queries in practice are conjunctive
- Some optimizers handle only conjunctive queries - break larger queries into many CQs
- CQ’s have “better” theoretical properties than arbitrary queries
Conjunctive Queries
in rule-based (datalog) notation

- **R**: Extensional database (EDB) - stored
- **P**: Intentional database (IDB) - computed

\[ P(x,z) \leftarrow R(x,y) \land R(y,z) \]

Variables

Head

Body

Subgoals

Implicit \( \exists \)

Conjunction

“IF”
Conjunctive Queries

Intuitively: when facts in the body are true of stored relations, then we infer the fact in the head

\[ P(x,z) :\quad R(x,y) \land R(y,z) \]

• More formally:
• Consider all possible substitutions: assignments of the variables in the body
Examples

EDB Relation: ManagedBy(emp,mgr)

A(x) :- ManagedBy(“Smith”,y) & ManagedBy(x,y)

All employees having the same manager as “Smith”

<table>
<thead>
<tr>
<th>A(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ManagedBy(x,y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sam</td>
</tr>
<tr>
<td>Smith</td>
</tr>
<tr>
<td>Sally</td>
</tr>
<tr>
<td>Smith</td>
</tr>
</tbody>
</table>

Substitution: y = Joan, x = Sally, x = Smith
Defining answers to CQ

- A substitution $v$ is a function from variables into the domain. e.g. $x \rightarrow a$, $y \rightarrow a$, $z \rightarrow b$, $u \rightarrow c$
- Let $I$ be an instance, i.e. relations $I(R_1) \ldots I(R_n)$
- A tuple $t$ is in the answer $q(I)$ if there is a substitution $v$ s.t:
  - $v(u_i) \in I(R_1)$ for each $i$, and
  - $t = v(u)$

`ans(u) :- R1(u1) & ... & Rn(un)`

**e.g.** $u_i = (x,y,z)$

$v(u_i) = (a,a,b)$
Examples

EDB Relation: ManagedBy(emp,mgr)

• Find all employees having the same director as Smith:

\[
A(x) \leftarrow \text{ManagedBy("Smith",y), ManagedBy(y,z), ManagedBy(x,u), ManagedBy(u,z)}
\]

(Your director is your manager’s manager)
Query language classes

Recursive Queries

FO queries

Conjunctive Queries

Algebra
Logic
SQL

RA
(safe) RC
SFW + UNION EXCEPT

RA:
σ, π, ×
single datalog rule
CQ and RA

Relational Algebra:
- CQ correspond precisely to $\sigma_C$, $\Pi_A$, $\times$
  (missing: $\cup$, $-$)

$A(x) :- \text{ManagedBy("Smith",y), ManagedBy(x,y)}$
Query language classes

Recursive Queries

FO queries

Conjunctive Queries

Algebra

Logic

SQL

RA

(safe) RC

SFW +

UNION

EXCEPT

RA: \( \sigma, \pi, \times \)

single datalog rule

S_dFW
CQ and SQL

Rule-based:

\[ A(x) :- \text{ManagedBy(“Smith”,y), ManagedBy(x,y)} \]

SQL:

```
select distinct m2.name
from ManagedBy m1, ManagedBy m2
where m1.name=“Smith” AND
    m1.manager=m2.manager
```
Boolean queries

A() :- ManagedBy(“Smith”, x), ManagedBy(“Sally”, x)

Is there someone who manages both Smith and Sally?

• Returns:
  – relation \{ ⟨⟩ \} if the answer is yes
  – relation \{ \} if the answer is no
Properties of Conjunctive Queries

• Satisfiability
  – A query q is **satisfiable** if there exists some input relation I such that q(I) is non-empty.
  – **FACT:** Every CQ is satisfiable.

• Monotonicity
  – A query q is **monotonic** if for each instance I,J over schema, I ⊆ J implies q(I) ⊆ q(J).
  – **FACT:** Every CQ is monotonic.
Satisfiability of CQs

We can always generate satisfying EDB relations from the body of the rule.

\[ S(x, y, z) :\neg P(x, w) \land R(w, y, v) \land P(v, z) \]

\[ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \]

\[ a \quad b \quad b \quad c \quad d \quad d \quad e \]

S \quad P \quad R
Monotonicity of CQs

Consider two databases I, J s.t. I ⊆ J.

let t ∈ q(I).

Then for some substitution v:

- v(ui) ∈ I(Ri) for each i.
- t = v(u)

Since I ⊆ J, v(ui) ∈ J(Ri) for each i

So t ∈ q(J)

ans(u) :- R1(u1) & ... & Rn(un)

e.g. ui = (x,y,z)
Consequence of monotonicity

Product (pname, price, category, maker)
Find products that are more expensive than all those produced by "Gizmo-Works"

```
SELECT name
FROM Product
WHERE price > ALL (SELECT price
                   FROM Purchase
                   WHERE maker='Gizmo-Works')
```

• This query is NOT monotone.
• Therefore, it is not in the class of conjunctive queries.
• It cannot be expressed as a simple SFW query.