Conjunctive Queries

• **Definition** A conjunctive query is defined by:

\[
\varphi := R(t_1, ..., t_{\text{arity}}) \quad | \quad t_i = t_j \quad | \quad \varphi \land \varphi' \quad | \quad \exists x . \varphi
\]

- missing are \( \forall, \lor, \lnot \)
- \( \text{CQ} \subseteq \text{FO} \)

Conjunctive Queries, CQ

• Example of CQ

\[
q(x,y) = \exists z . (R(x,z) \land \exists u . (R(z,u) \land R(u,y)))
\]

\[
q(x) = \exists z . \exists u . (R(x,z) \land R(z,u) \land R(u,y))
\]

• Examples of non-CQ:

\[
q(x,y) = \forall z . (R(x,z) \land R(y,z))
\]

\[
q(x) = T(x) \lor \exists z . S(x,z)
\]

Conjunctive Queries

• Any CQ query can be written as:

\[
q(x_1, ..., x_n) = \exists y_1 . \exists y_2 . ... \exists y_p . (R_1(t_11, ..., t_{1m}) \land ... \land R_k(t_{k1}, ..., t_{km}))
\]

(i.e. all quantifiers are at the beginning)

• Same in **Datalog** notation:

\[
\begin{align*}
\text{head} & : q(x_1, ..., x_n) \leftarrow R_1(t_{11}, ..., t_{1m}), ..., R_k(t_{k1}, ..., t_{km}) \\
\text{body} & : (i.e. all quantifiers are at the beginning)
\end{align*}
\]

Examples

Employee(x), ManagedBy(x,y), Manager(y)

• Find all employees having the same manager as “Smith”:

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

Employee(x), ManagedBy(x,y), Manager(y)

• Find all employees having the same director as Smith:

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

CQs are useful in practice

CQ and SQL

CQ:

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

SQL:

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

Notice “distinct”

CQ and RA

Relational Algebra:

• CQ correspond precisely to \( \sigma_C, \Pi_A, \times \)

(missing: \( \cup, – \))

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

Extensions of CQ

CQ⁺

Find managers that manage at least 2 employees

\[ A(y) :\text{-} ManagedBy(x,y), ManagedBy(z,y), x\neq y \]

Extensions of CQ

CQ⁺

Find employees earning more than their manager:

\[ A(y) :\text{-} ManagedBy(x,y), Salary(x,u), Salary(y,v), u>v \]
Extensions of CQ

**CQ**

Find people sharing the same office with Alice, but not the same manager:

\[ A(y) \gets \text{Office}("Alice", u), \text{Office}(y, u), \neg \text{ManagedBy}("Alice", x), \neg \text{ManagedBy}(x, y) \]

---

**UCQ**

Union of conjunctive queries

Datalog:

\[ A(\text{name}) \gets \text{Employee}(\text{name}, \text{dept}, \text{age}, \text{salary}), \text{age} > 50 \]
\[ A(\text{name}) \gets \text{RetiredEmployee}(\text{name}, \text{address}) \]

Datalog notation is very convenient at expressing unions (no need for \( \lor \))

---

Extensions of CQ

- If we extend too much, we capture FO

- Theoreticians need to be careful: small extensions may make a huge difference on certain theoretical properties of CQ

---

Query Equivalence and Containment

- Justified by optimization needs

- Intensively studied since 1977

---

Query Equivalence

- Queries \( q_1 \) and \( q_2 \) are **equivalent** if for every database \( D \), \( q_1(D) = q_2(D) \).

- Notation: \( q_1 \equiv q_2 \)

---

Query Equivalence

\[
\begin{align*}
\text{SELECT } & x.\text{name}, x.\text{manager} \\
\text{FROM } & \text{Employee } x, \text{Employee } y \\
\text{WHERE } & x.\text{dept} = 'Sales' \text{ and } x.\text{office} = y.\text{office} \\
& \text{and } x.\text{floor} = 5 \text{ and } y.\text{dept} = 'Sales'
\end{align*}
\]

Hmmmm… Is there a simple way to write that?
Query Containment

- Query $q_1$ is contained in $q_2$ if for every database $D$, $q_1(D) \subseteq q_2(D)$.
- Notation: $q_1 \subseteq q_2$
- Obviously: $q_1 \subseteq q_2$ and $q_2 \subseteq q_1$ iff $q_1 \equiv q_2$
- Conversely: $q_1 \land q_2 \equiv q_2$ iff $q_1 \subseteq q_2$

We will study the containment problem only.

Examples of Query Containments

- Is $q_1 \subseteq q_2$?
  - $q_1(x) : R(x,u), R(u,v), R(v,w)$
  - $q_2(x) : R(x,u), R(u,v), R(u,w)$

Examples of Query Containments

- Is $q_1 \subseteq q_2$?
  - $q_1(x) : R(x,u), R(u,v), R(v,w)$
  - $q_2(x) : R(x,u), R(u,v), R(u,w)$

Examples of Query Containments

- Is $q_1 \subseteq q_2$?
  - $q_1(x) : R(x,u), R(u,v), R(v,x)$
  - $q_2(x) : R(x,u), R(u,x)$

Examples of Query Containments

- Is $q_1 \subseteq q_2$?
  - $q_1(x) : R(x,u), R(u,v)$
  - $q_2(x) : R(x,u), R(u,v), R(v,w)$

Query Containment

- **Theorem** Query containment for FO is undecidable
- **Theorem** Query containment for CQ is decidable and NP-complete.
Query Containment Algorithm

How to check \( q_1 \subseteq q_2 \)

- **Canonical database** for \( q_1 \) is:
  \[ D_{q_1} = (D, R_{q_1}^1, \ldots, R_{q_1}^n) \]
  - \( D = \) all variables and constants in \( q_1 \)
  - \( R_{q_1}^1, \ldots, R_{q_1}^n = \) the body of \( q_1 \)

- **Canonical tuple** for \( q_1 \) is:
  \( \tau_{q_1} \) (the head of \( q_1 \))

Examples of Canonical Databases

\[ q_1(x,y) \iff R(x,u), R(u,v), R(v,y) \]

- Canonical database: \( D_{q_1} = (D, R^0) \)
  \[ \begin{array}{cc}
  x & u \\
  y & v \\
  
  \end{array} \]

- Canonical tuple: \( \tau_{q_1} = (x,y) \)

Examples of Canonical Databases

- Canonical database: \( D_{q_1} = (D, R) \)
  \[ \begin{array}{ccc}
  x & u & Smith \\
  u & v & Fred \\
  
  \end{array} \]

- Canonical tuple: \( \tau_{q_1} = (x) \)

Checking Containment

**Theorem:** \( q_1 \subseteq q_2 \iff \tau_{q_1} \in q_2(D_{q_1}) \).

Example:

\[ q_1(x,y) \iff R(x,u), R(u,v), R(v,y) \]

- \( D = \{x,u,v\} \)
- \( R = \begin{array}{cc}
  x & u \\
  v & y \\
  
  \end{array} \)

- Yes, \( q_1 \subseteq q_2 \)

Query Homomorphisms

- A **homomorphism** \( f : q_2 \rightarrow q_1 \) is a function \( f : \text{var}(q_2) \rightarrow \text{var}(q_1) \cup \text{const}(q_1) \)
  such that:
  - \( f(\text{body}(q_2)) \subseteq \text{body}(q_1) \)
  - \( f(\tau_{q_1}) = \tau_{q_2} \)

**The Homomorphism Theorem:** \( q_1 \subseteq q_2 \iff \) there exists a homomorphism \( f : q_2 \rightarrow q_1 \)

Example of Query Homeomorphism

\[ \begin{array}{c}
  \text{var}(q_1) = \{x, u, v, y\} \\
  \text{var}(q_2) = \{x, u, v, w, t, y\} \\
  
  \end{array} \]

\( q_1(x,y) \iff R(x,u), R(u,v), R(v,y) \)
\( q_2(x,y) \iff R(x,u), R(u,v), R(v,w), R(t,w), R(t,y) \)

Therefore \( q_1 \subseteq q_2 \)
Example of Query Homeomorphism

\[ \text{var}(q_1) \cup \text{const}(q_1) = \{x, u, "Smith"\} \]

\[ \text{var}(q_2) = \{x, u, v, w\} \]

\[ q_1(x) :- \ R(x, u), R(u,"Smith"), R(u,"Fred"), R(u, u) \]

\[ q_2(x) :- \ R(x, u), R(u, v), R(u,"Smith"), R(v, u) \]

Therefore \( q_1 \subseteq q_2 \)

The Homeomorphism Theorem

**Theorem** Conjunctive query containment is:

1. decidable (why?)
2. in NP (why?)
3. NP-hard

• Short: it is NP-complete

Query Containment for UCQ

\[ q_1 \cup q_2 \cup q_3 \cup \ldots \cup q_i \subseteq q_j \cup q_i \cup q_i \cup \ldots \]

Notice: \( q_1 \cup q_2 \cup q_3 \cup \ldots \subseteq q \) iff

\[ q_i \subseteq q \text{ and } q_j \subseteq q \text{ and } q_3 \subseteq q \text{ and } \ldots \]

**Theorem** \( q \subseteq q_i \cup q_j \cup q_i \cup \ldots \) iff there exists some \( k \) such that \( q \subseteq q_k \).

It follows that containment for UCQ is decidable, NP-complete.

Query Containment for CQ

\[ q_1(x) :- \ R(x, y), R(y, x) \]

\[ q_2(x) :- \ R(x, y), x < y \]

\( q_1 \subseteq q_2 \) although there is no homomorphism!

To check containment do this:

• Consider all possible orderings of variables in \( q_1 \)
• For each of them check containment of \( q_1 \) in \( q_2 \)
• If all hold, then \( q_1 \subseteq q_2 \)

Still decidable, but harder than NP: now in \( \Pi_2 \)