Extensions of CQs
Query language classes

Recursive Queries

FO queries

Conjunctive Queries

Expressiveness

RA

(safe) RC

SFW +
UNION EXCEPT

Conjunctive Queries: RA: \( \sigma, \pi, \chi \)

single datalog rule

S\(^d\)FW

RA, Logic, SQL

Algebra

Conjunctive Queries
Extensions of CQ: disequality

\[ CQ \neq \]

Find managers that manage at least 2 employees

\[ A(y) \ :- \ ManagedBy(x,y), \ ManagedBy(z,y), \ x \neq z \]
Extensions of CQ: inequality

\[ CQ^< \]

Find employees earning more than their manager

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

Additional EDB Relation: Salary(emp,money)
Extensions of CQ: negation

\[\text{CQ}^-\]

*Find people sharing the same office with Alice, but with a different manager*

\[
\text{A(y)} :\text{- Office(“Alice”,u), Office(y,u), ManagedBy(“Alice”,x), } \neg\text{ManagedBy(y,x)}
\]

Additional EDB Relation: Office(emp,officenum)
Extensions of CQ: union

UCQ
Unions of conjunctive queries

Rule-based:

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

Datalog notation is very convenient for expressing unions (no need for \( \lor \) )
Query language classes

Conjunctive Queries
- RA: $\sigma, \pi, \bowtie$
- single datalog rule
- S$^d$FW

Recursive Queries
- FO queries
- RA
- (safe) RC

Expressiveness

Algebra
- Logic
- SQL

SFW + UNION EXCEPT
- UCQ
- CQ$<$
- CQ$\neq$
- CQ$ightarrow$
Extensions of CQ

• If we extend too much, we capture FO
  – Namely: CQs + Union, Negation

• Theoreticians need to be careful: small extensions may make a huge difference on certain theoretical properties of CQ
Query language classes

Recursive Queries
FO queries

Conjunctive Queries

Expressiveness

Algebra Logic SQL

RA
(safe) RC

UCQ^-

UCQ
CQ<
CQ≠
CQ^-

RA: σ,π,×

single datalog rule

S^dFW

SFW + UNION EXCEPT
Query Equivalence and Containment

- One kind of static analysis
- Useful for query optimization
- Intensively studied since 1977
Query Equivalence

```sql
SELECT x.name, x.manager
FROM Employee x, Employee y
WHERE x.dept = 'Sales' and x.office = y.office
    and x.floor = 5 and y.dept = 'Sales'
```

Hmmmm…. Is there a simpler way to write that?
Query Equivalence

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

- Notation: $q_1 \equiv q_2$

relations equal
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.
Sidenote: containment for Boolean queries

- Recall: $q_1$ is **contained** in $q_2$ if for every database $D$, $q_1(D) \subseteq q_2(D)$.
  - if $q_1$, $q_2$ are boolean they return \{ ⟨⟩ \} or \{ ⟩ \}
  - containment says:
    - whenever $q_1(D) = \{ ⟨⟩ \}$ then $q_2(D) = \{ ⟨⟩ \}$.

- Containment is logical implication: $q_1 \rightarrow q_2$
Examples of Query Containments

Is $q_1 \subseteq q_2$?

$q_1(x) :- R(x,y), R(y,z), R(z,w)$

$q_2(x) :- R(x,y), R(y,z)$
Examples of Query Containments

Is $q_1 \subseteq q_2$?

$q_1(x) := R(x,y), R(y,z), R(z,x)$

$q_2(x) := R(x,y), R(y,x)$

Counter-example

Counter-example
Examples of Query Containments

Is $q_1 \subseteq q_2$?

$q_1(x) :- R(x,u), R(u,u)$

$q_2(x) :- R(x,u), R(u,v), R(v,w)$

Example
Examples of Query Containments

Is \( q_1 \subseteq q_2 \)?

\[
q_1(x) :- R(x,u), R(u,"Smith")
\]
\[
q_2(x) :- R(x,u), R(u,v)
\]
Query Containment

• **Theorem** Query containment for CQ is decidable and NP-complete.

(query complexity)
Checking containment

\[ q_1(x) :- R(x,u), R(u,u) \]
\[ q_2(x) :- R(x,u), R(u,v), R(v,w) \]

1. “Freeze” \( q_1 \)
   - Replace variables by unique constants
   - \( x \rightarrow a_x, u \rightarrow a_y \)
   - this is called canonical database of \( q_1 \)

2. Evaluate \( q_2 \) on frozen body of \( q_1 \)

3. If frozen head is derived, then \( q_1 \subseteq q_2 \)

\[ q_1(x) :- R(x,u), R(u,u) \]
\[ q_2(x) :- R(x,u), R(u,v), R(v,w) \]

Substitution:
\[ x = a_x, u = a_y, v = a_y, w = a_y \]

Containment!
Why does this test work?

- If the test is negative, the canonical database constructed is a counterexample to containment.
- If the test is positive:
  - substitution $v: \text{var}(q2) \rightarrow \text{“canonical domain”}$
  - this implies $f: \text{var}(q2) \rightarrow \text{var}(q1) \cup \text{const}(q1)$
  - Now suppose $t \in q1(I)$ for any instance $I$
    - there is substitution $w: \text{var}(q1) \rightarrow \text{domain}$
      - such that $t$ is derived.
    - then $f$ followed-by $w$ is a substitution showing that $t$ will be in $q2(I)$. 

44
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:
  - $f(\text{body}(q_2)) \subseteq \text{body}(q_1)$
  - $f(t_{q_1}) = t_{q_2}$

The Homomorphism Theorem $q_1 \subseteq q_2$ iff there exists a homomorphism $f : q_2 \rightarrow q_1$

Chandra & Merlin 1977
The Homomorphism Theorem

• **Theorem** Conjunctive query containment is:
  1. decidable (why ?)
  2. in NP (why ?)
  3. NP-hard

• In short: containment for CQs is NP-complete
Query Minimization

**Definition** A conjunctive query $q$ is **minimal** if for every other conjunctive query $q'$ s.t. $q \equiv q'$, $q'$ has at least as many predicates (‘subgoals’) as $q$.

Are these queries minimal?

```
q(x) :- R(x,y), R(y,z), R(x,x)
```

```
q(x) :- R(x,y), R(y,z), R(x,'Alice')
```
Query Minimization

• Query minimization algorithm

  Choose a subgoal $g$ of $q$
  Remove $g$: let $q'$ be the new query
  We already know $q \subseteq q'$ (why ?)
  If $q' \subseteq q$ then permanently remove $g$

• Notice: the order in which we inspect subgoals doesn’t matter
Other containment problems

• Extensions of CQs:
  – Unions of CQs
  – CQs with inequality

• FO queries
• Containment under constraints
• What about bags?
  – strange things happen
Containment under constraints

- Recall: query $q_1$ is **contained** in $q_2$ if for every database $D$, $q_1(D) \subseteq q_2(D)$.
- What if we know more about our input databases?
- Replace “every database $D$”, with:
  - “every database satisfying constraint $C$”

Containment under FD is NP-complete
Containment for FO queries

- **Theorem** Satisfiability for FO queries is undecidable
- **Lemma** Query containment/equivalence for FO is undecidable
  - if we had an algorithm for equivalence, we could use it to decide satisfiability of q:
  - check: \( q \equiv \text{false} \)

Consequence: we cannot do global query optimization for first-order queries.
Review

• CQs are an important fragment of FO
  – Equivalences: RA: $\sigma, \pi, \times$  SQL: $S_d^{FW}$
  – Properties: satisfiable, monotonic
  – containment/equivalence decidable, NPC

• Expressiveness
  – CQs strictly less expressive than FO

• Hardness of static optimization