Query Equivalence and Containment

Problem Definition

We want to determine whether two queries are the *same*, independent of *syntax*.

A query Q_1 is *contained* in Q_2 ($Q_1 \subseteq Q_2$) if:
For any database instance D, $Q_1(D) \subseteq Q_2(D)$.

The queries Q_1 and Q_2 are equivalent if

- $Q_1 \subseteq Q_2$ and
- $Q_2 \subseteq Q_1$.

Why Do We Care?

- Fundamentally interesting problem.
- Enables removing redundant subgoals.
- Validate query transformations.
- Detecting independence of queries from updates.
- Semantic caching.
- Answering queries using views.
- Knowledge-base verification.

Removing Redundant Subgoals

Consider the query:

$Q : q(X, Y) : -edge(Y, X), edge(Z, X), edge(X, Z_1).$

The second subgoal is redundant, and it is equivalent to:

$Q_1 : q_1(X, Y) : -edge(Y, X), edge(X, Z).$

Results Summary

Conjunctive queries: NP-complete, containment mappings.

Unions of CQs: still, NP-complete.

Built-in predicates: Π_2^p-complete; canonical databases.

Negation: more canonical databases.

More negation: undecidable.

Equivalence of datalog: undecidable.

Datalog in a CQ: decidable; triply exponential time;

Semi-structured data, OODB's, ...,

Containment Mappings

Q_1 contains Q_2 if and only if there is a containment mapping from Q_1 to Q_2 [Chandra and Merlin, 77].

$\phi : Variables(Q_1) \rightarrow Variables(Q_2)$ such that:

- for every atom L in the body of Q_1, $\phi(L)$ is in the body of Q_2, and
- ϕ maps the head of Q_1 to the head of Q_2.

$Q_1 : q_1(A, B) : -edge(B, A), edge(C, A), edge(A, D).$
$Q_2 : q_2(X, Y) : -edge(Y, X), edge(X, Z).$

The containment mapping is:
$\{A \rightarrow X,\ B, C \rightarrow Y, D \rightarrow Z\}.$

Independence of Queries from Updates

$$\begin{aligned} canDrive(X, Y, A) &:- \ inCar(X, Y, A),\ driver(X), \\ &\quad\ \ inCar(Z, Y, B),\ B \geq 18. \\ adultDriver(X) &:- \ canDrive(X, Y, A),\ A \geq 18. \end{aligned}$$

Deletion update: $inCar(X, Y, A),\ \neg driver(X)$.

$$\begin{aligned} canDrive'(X, Y, A) &:- \ inCar(X, Y, A),\ driver(X), \\ &\quad\ \ inCar(Z, Y, B),\ B \geq 18, \\ &\quad\ \ driver(Z). \\ adultDriver'(X) &:- \ canDrive'(X, Y, A),\ A \geq 18. \end{aligned}$$

Mathematical Logic Perspective

Containment is a special case of validity:

$Q : q(X, Y) : -edge(Y, X), edge(Z, X), edge(X, Z_1).$
contains
$Q_1 : q_1(X, Y) : -edge(Y, X), edge(X, Z)$
if and only if the following is valid:

$$\forall X, Y\ \exists Z(edge(Y, X) \wedge edge(X, Z)) \Rightarrow$$
$$\exists Z_1, Z_2(edge(Y, X) \wedge edge(Z_2, X) \wedge edge(X, Z_1)).$$

Decidability often follows from results in logic, but:

We contribute new complexity results, results on fixedpoint logic (recursive queries), and validity over finite models.

Canonical Databases

A different perspective on containment mappings (will be useful later).

To check whether Q_1 is contained in Q_2:

- Freeze the subgoals of Q_1: each subgoal becomes a tuple in the canonical database.
- Evaluate Q_2 on the canonical database.
- Containment holds iff the head of Q_1 is in the answer.

Evaluate $q_2(X, Y) : -edge(Y, X), edge(X, Z)$ on the database:

$\{edge(b, a), edge(c, a), edge(a, d)\}$.

Beyond Conjunctive Queries

Unions of conjunctive queries:

$q(A, B) : -edge(B, A), edge(C, A), edge(A, D)$.

$q(A, B) : -edge(B, A), edge(C, A), edge(B, C)$.

Q is contained in $Q_1 \vee \ldots \vee Q_n$ iff Q is contained in some Q_i (Sagiv & Yannakakis).

Conjunctive query in a datalog program: canonical databases work here too.

Datalog program in a conjunctive query: Decidably in doubly exponential time (Chaudhuri & Vardi).

Built-in Predicates

Containment mappings aren't enough:

$Q_1 : p(X, Y) : -e(X, Y), r(U, V), r(V, U)$

$Q_2 : p(X, Y) : -e(X, Y), r(U, V), U \leq V$

Q_1 is contained in Q_2, but there is no containment mapping.

Instead, you need to consider a set of canonical databases (Klug, 88).

In each canonical database, there is a complete ordering on the constants in the database.

Queries with Negation

Is $Q_1 \subseteq Q_2$?

$Q_1 : p(X, Z) : -a(X, Y), a(Y, Z), \neg a(X, Z)$
$Q_2 : p(A, C) : -a(A, B), a(B, C), \neg a(A, D)$

Need to consider all possible partitions of the variables in Q_1:

Case 1: $X \neq Y, X \neq Z, Y \neq Z$:
$\{a(0, 1), a(1, 2)\}$.

Case 2: $X = Y, X \neq Z$:
$\{a(0, 0), a(0, 1)\}$.

Evaluate Q_2 on each database, allowing for possibly more positive facts.