Answering Queries Using Views: The Last Frontier

The Problem

Given a query Q and a set of view definitions $V_1,\ldots,V_n$:
Is it possible to answer Q using only the V's?

V_1(A,B) :- cites(A,B), cites(B,A)
V_2(C,D) :- sameTopic(C,D), cites(C,C1), cites(D,D1)
Query:
q(x,y) :- sameTopic(x,y), cites(x,y), cites(y,x)

Query rewriting: q'(X,Y) :- V_1(X,Y), V_2(X,Y)
Unfolding of the rewriting:
q''(X,Y) :- cites(X,Y), cites(Y,X),
sameTopic(X,Y), cites(X,Z), cites(Y,W)

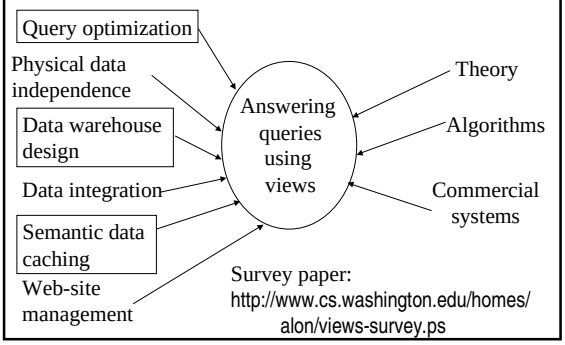
Another Example

French cars data source:
DB1(name, year) :- ForSale(name, year, "France", "auto"), year > 1990.
Car review database:
DB2(product, review) :- Review(product, review, "auto")

Query: q(X,Y,R):- ForSale(X,Y,C,"auto"), Review(X,R,"auto"), Y > 1985.
Query plan:
q'(X,Y,R) :- DB1(X,Y), DB2(X,R)

Note: rewriting is not equivalent to the query, but we can't do any better.

Motivation



Survey paper:
http://www.cs.washington.edu/homes/alon/views-survey.ps

Dimensions of the Problem

- View definition language
- Query language
- Semantic constraints (e.g., FD's, inclusions)
- Completeness/soundness of the views
- Output: query execution plan or logical plan.
- Equivalent or maximally contained rewriting.

Usability Conditions

Query: q(X,Z) :- r(X,Y), s(Y,Z), t(X,Z), Y > 5.

What can go wrong?

V1(A,B) :- r(A,C), s(C1,B) (join predicate not applied)
V2(A,B) :- r(A,C), s(C,B), C > 1 (predicate too weak).
V3(A,B) :- r(A,B), r1(A,B) (irrelevant condition).
V4(A) :- r(A,B), s(B,C), t(A,C), B > 5:
needed argument is projected out. Can be recovered if we have a functional dependency t: A --> C.

See [Larson & Yang, 87 and LMSS-95] for conditions.

Formal Definition: Rewriting

Given a query Q and a set of view definitions $V_1,\ldots,V_n$

Q' is a rewriting of the query using V's if it refers *only* to the views or to interpreted predicates.

Q' is an equivalent rewriting of Q using the V's if Q' is equivalent to Q.

Q' is a maximally-contained rewriting of Q w.r.t. ***L*** using the V's if there is no other Q'' such that:

Q'' strictly contains Q', and Q'' is contained in Q.

A Basic Decidability Result

- For conjunctive queries with no interpreted predicates, the following holds:
  - *If Q has an equivalent rewriting using V, then there exists one with no more conjuncts than Q.*
  [Levy, Mendelzon, Sagiv & Srivastava, PODS95]
- The rewriting problem is NP-complete.
- Bound holds even if views have interpreted predicates.
- Maximally-contained rewriting: union of all conjunctive rewritings of the length of the query or less.

Certain Answers

Given:
A query Q,
View definitions $V_1,\ldots V_n$,
Extensions of the views: $v_1,\ldots v_n$.

Consider the set of databases ***D*** that are consistent with $V_1,\ldots V_n$ and $v_1,\ldots v_n$.

The tuple *t* is a certain answer to Q if it would be an answer in every database in ***D.***

Note: an equivalent rewriting provides all certain answers.

Finding All Answers from Views

If a rewriting is equivalent: you definitely get all answers

Maximal containment: only w.r.t. a specific query language.
So what is the complexity of finding all the answers?

[Abiteboul & Duschka, PODS-98],
[Grahne and Mendelzon, ICDT-99]: *surprisingly hard!*

Certain answers:
Given specific extensions $v_1,\ldots v_n$ to the view, is the tuple *t* is an answer in *every* database *D*
that is consistent with the extensions $v_1,\ldots,v_n$?

Why & When is it Hard?

Sources can be:
• sound (open world assumption)
• complete
• sound and complete (closed-world assumption)

• If sources are either all sound or all complete, then maximally-contained rewriting exists.
• If the query contains interpreted predicates, the problem is NP-hard.
• If sources are sound and complete, the problem is NP-complete.

Graph Colorability as Views

- V1(X) :- edge(X,Y) (set of nodes in the graph)
- V2(Y) :- color(X,Z) (the set {red, green, blue})
- V3(X,Y):- edge(X,Y) (the set of edges).

Query:
- q(a) :- edge(X,Y), color(X,Z), color(Y,Z)

Potpourri

- System-R optimization extensions: [Tsatalos et al., VLDB94], Chaudhuri et al., ICDE-95].
- VLDB-98: Oracle's implemented algorithm.
- Infinite # of views [LRU, PODS-96, VP VLDB-97].
- Polynomial-time cases: [Chekuri & Rajaraman, ICDT-97].
- Description logics: [Calvanese et al. 99].
- Inclusion dependencies [Gryz, ICDE-97].
- Unions in views [Afrati et al, ICDT-99, Duscha's thesis].
- Semi-structured data: [VP, Sigmod-99].

Containment Queries over Views

[Millstein, Levy, Friedman, PODS-2000]

- Motivation: equivalence of queries to data integration systems.
- Two different queries can be equivalent given a specific set of sources.
- Certain(Q1) = Certain(Q2)?
- Π^{p}_{2} for the conjunctive query case.
- Is decidable in some cases where the maximally-contained rewriting is recursive.