ATFD Project : MiniCon Algorithm - A Scalable Algorithm for Answering Queries Using Views

Paridhi Mishra s1549106

University of Edinburgh

30th March 2016

Problem - Q, $V = V_1, \cdots, V_n$ over same schema, answer Q only in terms of views
Two contexts

- Query Optimization/Physical Data Independence - Query Equivalence
- Data Integration - Mediated Schema, Reformualtion, Maximally Contained Rewriting (instead of equivalent rewriting)

Problem: Conjunctive Query - set of Conjuntive views - Large no of views NP - complete Search space - exponential no fo rewritings
Conjunctive query - predicate, subgoals, head, body
$q(\bar{X}) : e_1(\bar{X}_1), \cdots, e_n(\bar{X}_n)$
$\bar{X}, \bar{X}_1, \cdots, \bar{X}_n$ have either variables or constants.
Variables $\bar{X}$ -*distinguished* , *existential* variables.

Containment mapping τ $Vars(Q_1) \to Vars(Q_2)$
maps(subgoal,head)
Query Containment $Q_1 \subseteq Q_2$ Q_2 contains Q_1 - mapping from Q_2 to Q_1
Query Equivalence $Q_1 \subseteq Q_2$ and $Q_2 \subseteq Q_1$
Query rewritings :

- *Equivalent rewritings:* Q' over $V_1(D), \cdots, V_n(D)$ and $Q(D)$.
- *Maximally-contained rewritings:* Given CQ , set of CV, L
  (1) $Q_1(v_1, v_2, \cdots, v_n) \subseteq Q(D)$ ($v_i \subseteq V_i(D)$, for $1 \leq i \leq n$)
  (2) NO query Q_2
    - $Q_2(v_1, v_2 \cdots, v_n) \subseteq Q(v_1, v_2 \cdots, v_n)$
    - $Q(v_1, v_2 \cdots, v_n) \subseteq Q(D)$

Previous Algorithms

- Search space - Maximally-contained rewriting
- Finite space - no comparison predicates in the query
- every possible conjunction of n or fewer view atoms, where n is the number of subgoals in the query.
- Restrictive space to produce rewritings faster
- Scalability issues (large no of views)- not much work done in literature

Bucket Algorithm : Subgoal considered separately - Bucket
Q1(x) :- cites(x,y),cites(y,x),sameTopic(x,y)
V4(a) :- cites(a,b), cites(b,a)
V5(c,d) :- sameTopic(c,d)
V6(f,h) :- cites(f,g),cites(g,h),sameTopic(f,g)
Step1: *Buckets* for each subgoal:

- *Bucket cites(x,y) = V4(x),V6(x,y)*
- *Bucket sameTopic(x,y) = V5(x,y),V6(x,y)*

Step2: Cartesian product - Conjunctive rewriting union - Containment check.

The Inverse-Rules Algorithm : Rules - Invert view definitions : *R1: cites(a, f1(a)) :- V4(a) R2: cites(f1(a), a) :- V4(a) R3: sameTopic(c,d) :- V5(c,d)*

MiniCon Algorithm
Phase 1

- Like bucket, looks views contain subgoals - query subgoals
- finds a partial mapping g query - g_1 in a view V
- looks at the variables - join predicates
- finds minimal additional set of subgoals that need to be mapped to subgoals in V

Subgoals + mapping information - **MiniCon Description**
MCD $(h_C, V(\bar{Y})_C, \phi_C, G_C)$
Phase 2-

- combine MCDs
- produce rewritings.
- no containment checks (advantange over bucket)

Contribution - Extension MiniCon algorithm- Schemas with functional dependencies

$student(S, P, Y), taught(P, D), program(P, C)$

Functional dependencies:

- $student : S \rightarrow P, S \rightarrow Y$
- $taught : P \rightarrow D$
- $program : P \rightarrow C$

$v1(S, Y, D) : -student(S, P, Y), taught(P, D)$
$v2(S, P) : -student(S, P, Y)$
$v3(P, C) : -program(P, C)$

Functional Dependency : A functional dependency $r : a_1, \cdots, a_n \rightarrow b$ in the mediated schema, where a1, ..., an and b refer to attributes in the relation r, states that for every two tuples t and u in r if $t.a_i = u.a_i$ for $i = 1, ..., n$, then $t.b = u.b$

Property 1. in Form MCD step for each MCD C

- **Clause 1.** For each head variable x of Q which is in the domain of ϕ_C, $\phi_C(x)$ is a head variable in $h_C(V)$
- **Clause 2.** If $\phi_C(x)$ is an existential variable in $h_C(V)$, then for every g, subgoal of Q, that includes x: (1) all the variables in g are in the domain of ϕ_C; and (2) $\phi_C(g) \in h_C(V)$

Property 2. in Combine MCD step for MCD C

The only combinations of MCDs that can result in *non-redundant rewritings* of Q are of the form $C_1, \cdots, C_l$, where:

- $G_{C_1} \cup \cdots \cup G_{C_l} = Subgoals(Q)$, and
- for every $i \neq j$, $G_{C_1} \cup \cdots \cup G_{C_j} = \emptyset$

MiniCon algorithm generates the following two rewritings only:
$q1(S,D):-v2(S,P),v5(P,D)$
$q3(S,D):-v6(S,D)$
Correct rewriting $query : q(S,P,Y):-v1(S,Y,D),v2(S,P)$
because the functional dependencies S $\rightarrow$ P and S $\rightarrow$ Y hold in the mediated schema.
When existing MinCon is implemented for this example

- MiniCon fails to generate the above rewritings.
- Not all the distinguished variables in the query subgoal can be mapped to the distinguished variables in v1
- So Clause C1 of Property 1 is violated
- No MCD for q over v1 can be used in a non-redundant rewriting of q.

Extension : Forming MCDs over Joint Views

- Construct a joint view v(1,2) of v1 and v2
- has all the distinguished variables in either v1 or v2 as its distinguished variables
- all the subgoalsof query in either v1 or v2 as its subgoals
- satisfies Clause C1 of Property 1
- generate MCD for q over v(1,2) covering the only subgoal in q
- non-redundant rewriting of q possible from MCD now

$V_1(S, Y, D) : -student(S, P, Y), taught(P, D)$
$V_2(S, P) : -student(S, P, Y)$
Joint view $V_{1,2}(S, Y, D, P) : -student(S, P, Y), taught(P, D)$
Implementation of extended algorithm FormMCD() in MiniCon is modified to create joint views and check for cases for Clause 1 of Property 1, allowing formation of MCD for functional dependency case.

Questions?