CS345 Notes for Lecture 10/14/96

Conjunctive Queries

= safe, datalog rules:

$$H \text{ :- } G_1 \,\&\, \cdots \,\&\, G_n$$

- Most common form of query; equivalent to select-project-join queries.
- Useful for optimization of active elements ("triggers," constraints, instantiated views).
- Useful for information integration.

Containment

$Q_1 \subseteq Q_2$ iff for every database D, $Q_1(D) \subseteq Q_2(D)$.

- Remember, $Q(D)$ is what we get by making all possible substitutions for variables of Q. If a substitution turns all subgoals of Q's body to facts in D, then the head of Q, with this substitution, is in $Q(D)$.
- Containment problem for CQ's is central. Problem is NP-complete, but not a "hard" problem in practical situations (short queries, few pairs of subgoals with same predicate).
- Function symbols do not make problems more difficult.
- Adding negated subgoals and/or arithmetic subgoals, e.g., $X < Y$, makes things more complex, but important special cases.

Example:

A: `p(X,Y) :- r(X,W) & b(W,Z) & r(Z,Y)`
B: `p(X,Y) :- r(X,W) & b(W,W) & r(W,Y)`

$B \subseteq A$. In proof, suppose $p(x, y)$ is in $B(D)$. Then there is some w such that $r(x, w)$, $b(w, w)$, and $r(w, y)$ are in D. In A, make the substitution

$$X \to x,\ Y \to y,\ W \to w,\ Z \to w$$

Thus, the head of A becomes $p(x, y)$, and all subgoals of A are in D. Thus, $p(x, y)$ is also in $A(D)$, proving $B \subseteq A$.

Testing Containment of CQ's

1. Containment mappings.
2. Canonical databases.

- Similar for basic CQ case, but (2) is useful for more general cases like negated subgoals.

Containment Mappings

Mapping from variables of CQ Q_2 to variables of CQ Q_1 such that

1. Head of Q_2 becomes head of Q_1.
2. Each subgoal of Q_2 becomes a subgoal of Q_1.

- It is not necessary that every subgoal of Q_1 is the target of some subgoal of Q_2.

Example: A, B as above. Containment mapping from A to B: $X \to X$, $Y \to Y$, $W \to W$, $Z \to W$.

- No containment mapping from B to A. Subgoal $b(W, W)$ in B can only go to $b(W, Z)$ in A. That would require both $W \to W$ and $W \to Z$.

Example:

C_1: `p(X) :- a(X,Y) & a(Y,Z) & a(Z,W)`
C_2: `p(X) :- a(X,Y) & a(Y,X)`

Containment mapping $C_1 \to C_2$:

$$X \to X,\ Y \to Y,\ Z \to X,\ W \to Y$$

- No containment mapping $C_2 \to C_1$. Proof:
  a) $X \to X$ required for head.
  b) Thus, first subgoal of C_2 must map to first subgoal of C_1; Y must map to Y.
  c) Similarly, 2nd subgoal of C_2 must map to 2nd subgoal of C_1, so X must map to Z.
  d) But we already found X maps to X.

Containment Mapping Theorem

$Q_1 \subseteq Q_2$ iff there exists a containment mapping from Q_2 to Q_1.

Proof (If)

Let μ: $Q_2 \rightarrow Q_1$ be a containment mapping. Let D be any DB.

- Every tuple t in $Q_1(D)$ is produced by some substitution σ on the variables of Q_1 that makes Q_1's subgoals all become facts in D.
- Claim: $\sigma \circ \mu$ is a substitution for variables of Q_2 that produces t.
    1. $\sigma \circ \mu(F_i) = \sigma(\text{some } G_j)$. Therefore, it is in D.
    2. $\sigma \circ \mu(H_2) = \sigma(H_1) = t$.
- Thus, every t in $Q_1(D)$ is also in $Q_2(D)$; i.e., $Q_1 \subseteq Q_2$.

Proof (Only If)

Key idea: *frozen* CQ.

1. Create a unique constant for each variable of the CQ Q.
2. Frozen Q is a database consisting of all the subgoals of Q, with the chosen constants substituted for variables.

Example:

```
p(X) :- a(X,Y) & a(Y,Z) & a(Z,W)
```

Let x be the constant for X, etc. The relation for predicate a consists of the three tuples (x, y), (y, z), and (z, w).

The proof: Let $Q_1 \subseteq Q_2$. Let database D be the frozen Q_1.

- $Q_1(D)$ contains t, the "frozen" head of Q_1 (sounds gruesome, but the reason is that we can use the substitution in which each variable of Q_1 is replaced by its corresponding constant).

- Since $Q_1 \subseteq Q_2$, $Q_2(D)$ must also contain t.

- Let σ be the substitution of constants from D for the variables of Q_2 that makes each subgoal of Q_2 a tuple of D and yields t as the head.

- Let σ' be the substitution that maps each variable X of Q_2 to the variable of Q_1 that corresponds to the constant $\sigma(X)$.

- σ' is a containment mapping from Q_2 to Q_1 because:

  a) The head of Q_2 is mapped by σ to t, and t is the frozen head of Q_1, so σ' maps the head of Q_2 to the "unfrozen" t, that is, the head of Q_1.

  b) Each subgoal F_i of Q_2 is mapped by σ to some tuple of D, which is a frozen version of some subgoal G_j of Q_1. Then σ' maps F_i to the unfrozen tuple, that is, to G_j itself.

Dual View of Containment Mappings

A containment mapping, defined as a mapping on variables, induces a mapping on subgoals.

- Therefore, we can alternatively define a containment mapping as a function on subgoals, thus inducing a mapping on variables.

- The containment mapping condition becomes: the subgoal mapping does not cause a variable to be mapped to two different variables or constants, nor cause a constant to be mapped to a variable or a constant other than itself.

Example: Again consider

A: `p(X,Y) :- r(X,W) & b(W,Z) & r(Z,Y)`
B: `p(X,Y) :- r(X,W) & b(W,W) & r(W,Y)`

Previously, we found the containment mapping $X \to X$, $Y \to Y$, $W \to W$, $Z \to W$ from A to B.

- We could as well describe this mapping as $r(X,W) \rightarrow r(X,W)$, $b(W,Z) \rightarrow b(W,W)$, and $r(Z,Y) \rightarrow r(W,Y)$.

Method of Canonical Databases

Instead of looking for a containment mapping from Q_2 to Q_1 in order to test $Q_1 \subseteq Q_2$, we can apply the following test:

1. Create a *canonical* database D that is the frozen body of Q_1.
2. Compute $Q_2(D)$.
3. If $Q_2(D)$ contains the frozen head of Q_1, then $Q_1 \subseteq Q_2$; else not.

- The proof that this method works is essentially the same as the argument for containment mappings.
  - □ The only way the frozen head of Q_1 can be in $Q_2(D)$ is for there to be a containment mapping $Q_2 \rightarrow Q_1$.

Example:

C_1: `p(X) :- a(X,Y) & a(Y,Z) & a(Z,W)`
C_2: `p(X) :- a(X,Y) & a(Y,X)`

- Test $C_2 \subseteq C_1$.
- Choose constants $X \rightarrow 0$, $Y \rightarrow 1$.
- Canonical DB from C_1 is

$$D = \{a(0,1),\ a(1,0)\}$$

- $C_1(D) = \{p(0), p(1)\}$.
- Since the frozen head of C_2 is $p(0)$, which is in $C_1(D)$, we conclude $C_2 \subseteq C_1$.
- Note that the instantiation of C_1 that shows $p(0)$ is in $C_1(D)$ is $X \rightarrow 0$, $Y \rightarrow 1$, $Z \rightarrow 0$, and $W \rightarrow 1$.
  - □ If we replace 0 and 1 by the variables X and Y they stand for, we have the containment mapping from C_1 to C_2.