CQ's With Negation

General form of conjunctive query with negation (CQN):

$$H \texttt{ :- } G_1 \texttt{ \& ... \& } G_n \texttt{ \& } \\ \texttt{NOT } F_1 \texttt{ \& ... \& NOT } F_m$$

- G's are *positive* subgoals; F's are *negative* subgoals.
- Apply CQN Q to DB D by considering all possible substitutions of constants for the variables of Q. If for some substitution:

  1. *All* the positive subgoals become facts in D and
  2. *None* of the negative subgoals do,

  then infer the substituted head.
- Set of inferred facts is $Q(D)$.
- Containment of CQ's doesn't change: $Q_1 \subseteq Q_2$ iff for every database D, $Q_1(D) \subseteq Q_2(D)$.

Example

```
C1: p(X,Z) :- a(X,Y) & a(Y,Z) &
              NOT a(X,Z)
C2: p(A,C) :- a(A,B) & a(B,C) &
              NOT a(A,D)
```

- Intuitively, C_1 looks for paths of length 2 that are not "short-circuited" by a single arc from beginning to end.
- C_2 looks for paths of length 2 that start from a node A that is not a "universal source"; i.e., there is at least one node D not reachable from A by an arc.
- We thus expect $C_1 \subseteq C_2$, but not vice-versa.

Levy-Sagiv Test

There is a straightforward, time-consuming test for $Q_1 \subseteq Q_2$:

- Create a large-but-finite family of canonical DB's that consist of all DB's using only the constants $1, 2, \ldots, n$, where n is the number of variables in Q_1.

* Test each canonical DB. If $Q_1(D)$ is not contained in $Q_2(D)$ for even one canonical DB D, then containment of CQ's surely doesn't hold. Otherwise, we claim that $Q_1 \subseteq Q_2$.

Proof of L/S Test

* Suppose $Q_1(D) \subseteq Q_2(D)$ for each canonical DB D, but there is some other DB E, for which containment doesn't hold. That is, $Q_1(E)$ contains a tuple t that $Q_2(E)$ does not contain.

* Consider the at most n symbols that variables of Q_1 map to when showing that $Q_1(E)$ contains t. We may rename these symbols $1, 2, \ldots, n$; the counterexample still holds.

* Let D be the canonical DB consisting of E restricted to the tuples having only the symbols $1, 2, \ldots, n$.

* Since the L/S test passed, we know that $Q_2(D)$ contains t.

* Since the assignment of Q_2's variables that shows t is in $Q_2(D)$ maps variables only to $1, 2, \ldots, n$ (remember all CQ's are assumed safe), the same assignment maps the positive subgoals of Q_2 to tuples of E and negative subgoals of Q_2 to tuples not in E.

    ❖ In proof: note that D and E, after renaming of symbols, agree on all tuples that involve only $1, 2, \ldots, n$. That is, D and E "look the same" whenever we assign variables to only $1, 2, \ldots, n$.

CQ's With Arithmetic

Suppose we allow subgoals with $<$, $\neq$, and other comparison operators.

* We must assume database constants can be compared.

* Technique is a generalization of the L/S algorithm, but it is due to Tony Klug.

* We shall work the case where $<$ is a total order; other assumptions lead to other algorithms, and we shall later give an all-purpose technique using a different approach.

Example

Consider the rules:

C_1: `p(X,Z) :- a(X,Y) & a(Y,Z) & X<Y`
C_2: `p(A,C) :- a(A,B) & a(B,C) & A<C`

- Both ask for paths of length 2. But Q_1 requires that the first node be numerically less than the second, while Q_2 requires that the first node be numerically less than the third.

Klug/Levy/Sagiv Test

Construct a family of canonical databases by considering all partitions of the variables of Q_1 (assuming we are testing $Q_1 \subseteq Q_2$), and ordering the partitions.

- To represent canonical DB's assign the first partition the value 0, the second the value 1, and so on.

Example

To test $C_1 \subseteq C_2$:

C_1: `p(X,Z) :- a(X,Y) & a(Y,Z) & X<Y`
C_2: `p(A,C) :- a(A,B) & a(B,C) & A<C`

we need to consider the partitions of $\{X, Y, Z\}$ and order them.

- The number of ordered partitions is 13.
  - For partition $\{X\}\{Y\}\{Z\}$ we have $3! = 6$ possible orders of the blocks.
  - For the three partitions that group two variables and leave the other separate we have 2 different orders.
  - For the partition that groups all three, there is one order.

- In this example, the containment test fails. We have only to find one of the 13 cases to show failure.
- For instance, consider $\{X, Z\}\{Y\}$. The canonical database D for this case is $\{a(0,1), a(1,0)\}$, and since $X < Y$, the body of C_1 is true.
- Thus, $C_1(D)$ includes $p(0,0)$, the frozen head of C_1.

- However, no assignment of values to A, B, and C makes all three subgoals of C_2 true, when D is the database.
- Thus, $p(0,0)$ is not in $C_2(D)$, and D is a counterexample to $C_1 \subseteq C_2$.

Key Theorems No Longer Hold When Some Predicates are Interpreted (e.g., Arithmetic Comparisons)

- Union of CQ's theorem is false.

Example

Consider something we've seen before:

Q_1: `p(X) :- a(X) & 10≤X & X≤20`
R_1: `p(X) :- a(X) & 5≤X & X≤15`
R_2: `p(X) :- a(X) & 15≤X & X≤25`

$Q_1 \subseteq R_1 \cup R_2$, but neither $Q_1 \subseteq R_1$ nor $Q_1 \subseteq R_2$ is true.

- Containment mapping theorem is false.

Example

Q_1: `panic :- r(U,V) & r(V,U)`
Q_2: `panic :- r(U,V) & U≤V`

- Note, "panic" is a 0-ary predicate, i.e., a propositional variable.
  - ❖ 0-ary predicates in the head present no problems for CQ's but don't make anything easier either.
- Informally: Q_1 = "cycle of length 2"; Q_2 = "nondecreasing arc."
- Thus, $Q_1 \subseteq Q_2$.
  - ❖ That is, whenever there is a pair of arcs $U \rightarrow V$ and $V \rightarrow U$, surely one is nondecreasing.
- However, if μ is a containment mapping from Q_2 to Q_1, there is no subgoal that $\mu(U \leq V)$ can be.
- Hence, no containment mapping from Q_2 to Q_1.

Generalizing the Containment-Mapping Theorem

- The Klug/Levy/Sagiv approach uses canonical databases to handle arithmetic.
- Another approach, due to Ashish Gupta and Zhang/Ozsoyoglu, uses containment mappings.
  - ❖ It has the advantage of working for any kind of interpreted ("built-in") predicate, although we shall use arithmetic comparisons in our examples.

The G/Z/O Test

To test whether $Q_1 \subseteq Q_2$, where Q_1, Q_2 are CQ's with interpreted predicates:

1. *Rectification*: replace variables and constants by new variables so that no variable appears twice among the relational subgoals and the head. Also, no constant may appear there at all.
2. Add equality comparisons so the new variables are equated to the variable or constant they replace.

Examples

a) Q_1 above:

```
panic :- r(U,V) & r(V,U)
```

becomes

```
panic :- r(U,V) & r(X,Y) &
         U=Y & V=X
```

b)

```
p(X) :- q(X,Y,X) & r(Y,a)
```

would become:

```
p(Z) :- q(X,Y,W) & r(V,U) &
        X=W & X=Z & Y=V & U=a
```

G/Z/O Test (Continued)

3. Having modified the CQ's, let M be the set of all containment mappings from the relational subgoals of Q_2 to the relational subgoals of Q_1.
   - ❖ Note that with all variables appearing only once, every mapping from subgoals to subgoals that matches predicates gives us a containment mapping.

- Then $Q_1 \subseteq Q_2$ iff the interpreted subgoals of Q_1 logically imply the OR, over all μ in M, of μ applied to the interpreted subgoals of Q_2.

Example

Let

Q_1: `panic :- r(U,V) & r(X,Y) &`
`U=Y & V=X`
Q_2: `panic :- r(U,V) & U≤V`

- Two containment mappings:
  1. $\mu_1(U) = U$; $\mu_1(V) = V$. Here, the $r(U, V)$ subgoal of Q_2 maps to the first subgoal of Q_1.
  2. $\mu_2(U) = X$; $\mu_2(V) = Y$. Here, $r(U, V)$ of Q_2 maps to the second subgoal of Q_1.
- We must check:

$$U = Y \;\wedge\; V = X \;\Rightarrow\; \mu_1(U \le V) \;\vee\; \mu_2(U \le V)$$

That is:

$$U = Y \;\wedge\; V = X \;\Rightarrow\; U \le V \;\vee\; X \le Y$$

- Use equalities $U = Y$ and $V = X$ in the hypothesis. Sufficient to show:

$$U \le V \;\vee\; V \le U$$

(Obviously true).

Test For Logical Expressions Involving Inequalities

- For arbitrary interpreted predicates, we can only make the necessary test by using whatever algorithm is appropriate for those predicates.
- For interpreted predicates that are arithmetic inequalities, we can use the same test that was hidden inside the K/L/S test:
  - ❖ Consider all total orders of variables, including those with equalities.
- If implication holds for each order, then expression is true, else false.

Example

For the implication above:

$$U = Y \;\wedge\; V = X \;\Rightarrow\; U \le V \;\vee\; X \le Y$$

two possible orders are:

$U < V < X < Y$
$X < U = V < Y$

- For this implication, the only orders that make the hypothesis $(U = Y \wedge V = X)$ true are:

  $U = V = X = Y$
  $U = Y < V = X$
  $V = X < U = Y$

- Conclusion $U \leq V \vee X \leq Y$ holds for each of the three orders.

- Test is exponential but works.

Extensions

- Extends to test for a CQ contained in a union of CQ's. The logical implication includes the OR over all containment mappings from any of the CQ's in the union.

- Extends to containment of unions of CQ's: handle each CQ in the contained unions separately.