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:

- Consider all databases D for the predicates mentioned in the body of Q_1, that contain
  1. No more tuples for a predicate than Q_1 has positive subgoals with that predicate.
  2. Only tuples made from the symbols $1, 2, \ldots, n$, where n is the number of variables in Q_1.
- Test $Q_1(D) \subseteq Q_2(D)$ for all such D; say "yes" if all containments hold.
- Number of D's is finite, and if there is a counterexample, then one of the D's will be a counterexample (why?).

More Organized Levy/Sagiv Test

To test $Q_1 \subseteq Q_2$:

1. Construct the set of *basic canonical* databases that correspond to all the partitions of the set of variables of Q_1.

   - That is, for each partition, assign a unique constant to each block of the partition.

   - Create the basic canonical DB by replacing each variable by the constant of its block. The basic canonical DB is the set of resulting *positive* subgoals.

2. For each basic canonical DB D constructed in (1), check that:

   - If $Q_1(D)$ contains the frozen head of Q_1, then so does $Q_2(D)$.

   Note that unlike ordinary CQ's, it is possible that $Q_1(D)$ does not contain Q_1's head, because D may make a negated subgoal false (i.e., D contains the frozen subgoal without the `NOT`).

3. If $Q_1(D)$ contains the frozen head of Q_1, we must then also consider the larger set of (*extended*) canonical DB's D' formed by adding to D other tuples that are formed from the same symbols as D, but not any of the tuples that are the negated subgoals of Q_1.

   - Check that $Q_2(D')$ contains the frozen head of Q_1 (with respect to D).

4. If so for all D', then $Q_1 \subseteq Q_2$; if not, then not.

Example

Consider C_1 and C_2, as above:

C_1:
```
p(X,Z) :- a(X,Y) & a(Y,Z) &
          NOT a(X,Z)
```
C_2:
```
p(A,C) :- a(A,B) & a(B,C) &
          NOT a(A,D)
```

- The variables of C_1 are $\{X, Y, Z\}$.
- There are five partitions of the variables, shown in the table below.

	Partition	Basic Canonical DB D
1)	$\{X\}\{Y\}\{Z\}$	$\{a(0,1), a(1,2)\}$
2)	$\{X,Y\}\{Z\}$	$\{a(0,0), a(0,1)\}$
3)	$\{X\}\{Y,Z\}$	$\{a(0,1), a(1,1)\}$
4)	$\{X,Z\}\{Y\}$	$\{a(0,1), a(1,0)\}$
5)	$\{X,Y,Z\}$	$\{a(0,0)\}$

- In cases (2), (3), and (5), $C_1(D)$ does not contain its own frozen head.

    - E.g., in case (2), the frozen head $p(X,Z)$ is $p(0,1)$. But if $X = 0$ and $Z = 1$, then the subgoal `NOT a(X,Z)` must be false, because $a(0,1)$ is in D.

- In cases (1) and (4), $C_1(D)$ contains C_1's frozen head, but so does $C_2(D)$ and any extended canonical DB $D' \supseteq D$.

    - E.g., in case (4), the frozen head of C_1 is $p(0,0)$. $C_2(D)$ contains $p(0,0)$, as we can see from the substitution $A \rightarrow 0$, $B \rightarrow 1$, $C \rightarrow 0$, $D \rightarrow 0$.

    - Moreover, adding tuples consisting of 0's and 1's to the basic canonical database $\{a(0,1),\ a(1,0)\}$ cannot create a counterexample, since we are not allowed to add $a(0,0)$, the frozen negative subgoal of C_1.

    - For the only possible $D' = \{a(0,1),\ a(1,0),\ a(1,1)\}$, $C_2(D')$ contains C_1's frozen head: $p(0,0)$.

Example

Consider a slightly different pair of CQ's:

C_1: `p(X,Z) :- a(X,Y) & a(Y,Z) &`
`NOT a(X,Z)`

C_3: `p(A,C) :- a(A,B) & a(B,C) &`
`NOT a(C,C)`

- C_1 is as before, so the basic canonical DB's are also as in the previous example.
- However, consider the partition $\{X\}\{Y\}\{Z\}$.
- While for the resulting basic canonical DB $D = \{a(0,1), a(1,2)\}$, both $C_1(D)$ and $C_3(D)$ contain C_1's frozen head $p(0,2)$, the same is not true for the extended canonical DB $D' = \{a(0,1), a(1,2), a(2,2)\}$.

Proof L/S Algorithm Works: Only-If

Suppose $Q_1 \subseteq Q_2$. Let D be any of the canonical DB's.

- If $Q_1(D)$ contains the frozen head of Q_1, then since $Q_1 \subseteq Q_2$, so does $Q_2(D)$.
- Therefore, the L/S test will be positive for all canonical DB's.

Proof of L/S: If

Assume that for every canonical DB, the L/S test is positive. Let D be some DB, and suppose $Q_1(D)$ contains tuple t. We must show $Q_2(D)$ contains t.

- Let σ be the substitution for the variables of Q_1 that yields t.
  - I.e., for every positive subgoal G of Q_1, $\sigma(G)$ is in D, and for no negative subgoal F of Q_1 is $\sigma(F)$ in D. Also, $\sigma(H_1) = t$ (H_1 is the head of Q_1).
- Partition the variables of Q_1 according to the value σ assigns them.
- This partition yields a basic canonical DB, C.
  - Remember, C consists of the positive subgoals of Q_1, with each variable X replaced by $\sigma(X)$.
  - Thus, $C \subseteq D$.

- It must be that $Q_1(C)$ contains t.
  - ❖ Each positive subgoal of Q_1 is mapped by σ to a member of D, and this ground atom is also a member of C.
  - ❖ Each negative subgoal of Q_1 is mapped by σ to something not in D. Since $C \subseteq D$, no negative subgoal is mapped to a member of C.
- Since $Q_1(C)$ contains t, the L/S test must have considered every extended canonical database C' containing C.
  - ❖ Technically, the test may have renamed the constants of C in some one-one way, but we may assume the constants based on D's constants were used, for convenience.

- Apparently, all such tests were positive; i.e., if $Q_1(C')$ contains t (the frozen head H_1), then so does $Q_2(C')$.

- Let C' be the particular superset of C that is D, restricted to tuples that involve only constants that are in the range of σ.

- If we apply Q_1, with the mapping σ, to C', it must be that $Q_1(C')$ contains t, because C' "looks like" D as far as this particular mapping σ is concerned.

  - Thus, C' cannot contain a frozen negative subgoal of Q_1.

- Thus, $Q_2(C')$ also contains t, since the L/S test, which was successful, required us to test C' in this way.

- If we compute $Q_2(D)$, we also get t, because we can use the same mapping as we did when we computed $Q_2(C')$.

- The fact that $Q_2(D)$ contains t completes the proof.

CQ's With Arithmetic

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

- Negated subgoals also permitted.
- 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

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

- However, now we need to consider also the order of the values we assign to each partition.
- And if there is negation, we also need to consider extended canonical DB's; without negation, the basic canonical DB's are sufficient.

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 again need to consider the partitions of $\{X, Y, Z\}$. But now, order of the values counts too.

- The number of different basic canonical databases 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 $X = Z = 0$ and $Y = 1$. 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 $C_1 \subseteq C_2$.