Homework 2 - Solutions

A: SIZE BOUNDS FOR JOINS [25%]

1. [15%] Compute the maximum possible output size for the following queries. Assume that all relations have the same size $N$.

   (a) $q(x, y) : -R(x), S(x, y), T(y)$.
   (b) $q(x, y, z, w) : -R(x, y), S(y, z), T(z, w), U(w, x), V(x, z)$.
   (c) $q(x, y, z, w) : -R(x, y, z), S(x, z, w), T(x, y, w), U(y, z, w)$.

   Solution. For (a) the maximum size is $N$, for (b) it is $N^2$ and for (c) it is $N^{4/3}$.

2. [10%] Suppose that relation $R_i$ has size $N_i$. Compute the maximum output size for the following query. (Hint: there will be different cases depending on the specific $N_i$).

   $q(x_1, x_2, x_3, x_4) : -R_1(x_1, x_2), R_2(x_2, x_3), R_3(x_3, x_4), R_4(x_4, x_1)$.

   Solution. The maximum possible size is $\min\{N_1N_3, N_2N_4\}$.

B: DATALOG [75%]

1. [15%] We are given a directed graph represented by the binary relation $R(x, y)$. Write a Datalog program that computes the following unary relations:

   (a) $OddCycle(x)$: there exists a cycle of odd length that goes through node $x$,
   (b) $EvenCycle(x)$: there exists a cycle of even length that goes through node $x$.
   (c) $Cycle(x)$: there exists a cycle that goes through node $x$.

   Solution. The following program computes all relations:

   OddPath(x, y) :- R(x, y).
   EvenPath(x, y) :- R(x, z), OddPath(z, y).
   OddPath(x, y) :- R(x, z), EvenPath(z, y).
   OddCycle(x) :- OddPath(x, x).
   EvenCycle(x) :- EvenPath(x, x).
   Cycle(x) :- OddCycle(x).
   Cycle(x) :- EvenCycle(x).
2. [10%] Show that every Datalog query is monotone.

3. [10%] Is the following Datalog program equivalent to a UCQ query? If so, write the query. If not, prove why it is not the case.

\[ B(X, Y) : - L(X, Y). \]
\[ B(X, Y) : - T(X), B(Z, Y). \]

**Solution.** The UCQ \( B(x, y) = q_1 \cup q_2 \), where \( q_1(x, y) = T(x), L(z, y) \) and \( q_2(x, y) = L(x, y) \) is equivalent.

4. [10%] Consider the following Datalog program:

\[ T(x, y) :- F(x, y). \]
\[ T(x, y) :- up(x, z1), T(z1, z2), down(, y). \]

What is the immediate consequence operator for semi-naive evaluation? Run the semi-naive evaluation on the following instance:

\[ F(c, g), F(c, c), up(f, e), up(e, d), up(d, c), down(c, d), down(c, e), down(e, h). \]

**Solution.** The program will produce first \( (c, c), (c, g) \), the next iteration \( (d, d), (d, e) \) and finally \( (e, h) \).

5. [20%] Perform the magic set transformation for the following Datalog program:

\[ T(x, y) :- F(x, y). \]
\[ T(x, y) :- up(x, z1), T(z1, z2), F(z2, z3), T(z3, z4), down(z4, y). \]
\[ q(y) :- T(a, y). \]

6. [10%] Find all possible stratifications for the following Datalog program with negation:

\[ T(x) :- S(x), \text{not } R(x). \]
\[ S(x) :- T(x), \text{not } R(x). \]
\[ U(x) :- R(x), \text{not } T(x), \text{not } S(x). \]
\[ V(x, y) :- V(x, y), \text{not } U(x), \text{not } U(y), . \]

**Solution.** The only stratification is \{S, T\}, \{U\}, \{V\}. 

---

2