3. Evaluation and Containment of Conjunctive Queries

These are sample solutions to some of the exercises that were given as coursework. They are not intended as models but show each one way to approach the problem set in the exercise.

These sample solutions have been authored by Elena Botoeva, for the edition of this course in 2013/14.

1. Evaluation of Special Types of Conjunctive Graph Queries

1. The k-th circle query $C_k(x) :\!-\ \mathit{edge}(x, z_1), \mathit{edge}(z_1, z_2), \ldots, \mathit{edge}(z_{k-1}, x)$.

   The algorithm:

   ```
   $R := \mathit{edge}$
   **for** $i := 1$ **to** $k - 1$ **do**
       $S := R \bowtie_{R.2=\mathit{edge}.1} \mathit{edge}$
       $R := \pi_{S.1,S.3}(S)$
   **return** $\pi_{R.1}(\sigma_{R.1=R.2}(R))$
   ```

   Let $n = |\mathit{edge}^{\mathbf{I}}|$. The running time:

   - line 1: R is of size n.
   - line 3 (for $i = 1$): the join can be computed by sorting R on the second component and edge on the first component: $O(n \log(n))$. The resulting relation S is of size $O(n^2)$.
   - line 4 (for $i = 1$): the projection requires duplicate elimination, it can be done by sorting S on the first component: $O(n^2 \log(n^2)) = O(n^2 \log(n))$. The resulting relation R is of size $O(n^2)$.
   - line 3 (for $i = 2, \ldots$): now, R is of size $O(n^2)$, so the running time is $O(n^2 \log(n))$. The resulting relation S is of size $O(n^3)$.
   - line 4 (for $i = 2, \ldots$): now, S is of size $O(n^3)$, so the running time is $O(n^3 \log(n))$. The resulting relation R is of size $O(n^2)$.
   - line 5: selection can be done in liner time in the size of R, so $O(n^2)$, then projection requires sorting, so $O(n^2 \log(n))$.
   - the total running time is $O(kn^3 \log(n))$.

2. The k-th star query $S_k(x) :- \mathit{edge}(x, z_1), \mathit{edge}(x, z_2), \ldots, \mathit{edge}(x, z_k)$.

   Note that $S_k(x)$ is equivalent to the query $S'_k(x) :- \mathit{edge}(x, z_1)$.

   The algorithm:

   ```
   return $\pi_{\mathit{edge}.1}(\mathit{edge})$
   ```

   The running time is $O(n \log(n))$.

3. The k-th spider-web query $W_k(x) :- \begin{array}{l} \mathit{edge}(x, z_1), \mathit{edge}(x, z_2), \ldots, \mathit{edge}(x, z_k), \\ \mathit{edge}(z_1, z_2), \mathit{edge}(z_2, z_3), \ldots, \mathit{edge}(z_k, z_1). \end{array}$

   The algorithm:

   ```
   $R := \mathit{edge},\ \mathit{Chord} := \mathit{edge}$
   $S := R \bowtie_{R.2=\mathit{Chord}.1} \mathit{Chord}$          $//S$ is a ternary relation
   $T := S \bowtie_{S.1=R.1,S.3=R.2} R$          $//T$ is a ternary relation
   for $i := 1$ to $k - 1$ do
       $Q := T \bowtie_{T.3=\mathit{Chord}.1} \mathit{Chord}$          $//Q$ is a quaternary relation
       $S := \pi_{Q.1,Q.2,Q.4}(Q)$          $//S$ is a ternary relation
       $T := S \bowtie_{S.1=R.1,S.3=R.2} R$          $//T$ is a ternary relation
   return $\pi_{T.1}(\sigma_{T.2=T.3}(T))$
   ```

   The running time: the principal difference with the analysis for circle queries is that here we have a quaternary relation Q of size $O(n^4)$. So line 6 is the most expensive one and costs $O(n^4 \log(n))$. The total cost is $O(kn^4 \log(n))$.

4. The k-th clique query $Cl_k() :- \begin{array}{l} \mathit{edge}(z_1, z_2), \mathit{edge}(z_1, z_3), \ldots, \mathit{edge}(z_1, z_k), \\ \mathit{edge}(z_2, z_1), \mathit{edge}(z_2, z_3), \ldots, \mathit{edge}(z_2, z_k), \\ \ldots \\ \mathit{edge}(z_k, z_1), \mathit{edge}(z_k, z_2), \ldots, \mathit{edge}(z_k, z_{k-1}). \end{array}$

   The query $Cl_k()$ evaluates to true over $\mathbf{I}$ if and only if the graph defined by $\mathit{edge}^{\mathbf{I}}$ contains a clique of size smaller or equal k. The latter problem is known to be NP-complete. It means that the most efficient (known) algorithm for evaluating Cl_k runs in exponential time in the worst case scenario. The following is a brute-force algorithm:

   ```
   $S := \mathit{edge}^{\mathbf{I}} \cap (\mathit{edge}^{\mathbf{I}})^{-}$
   $V := \pi_{S.1}(S)$
   for each tuple $(x_1, \ldots, x_k) \in V^k$
       check whether $x_1, \ldots, x_k$ form a clique, i.e.,
           $(x_i, x_j) \in S,$ for all $i = 1, \ldots, k-1,\ j = i+1, \ldots, k$ and $x_i \neq x_j$
       if they do, return true
   return false
   ```

   The running time:

   - line 2: V is of size $O(n)$.
   - line 3: V^k is of size $O(n^k)$.
   - line 4: checking whether $x_1, \ldots, x_k$ form a clique can be done in $O(k^2)$.
   - the total cost is $O(n^k k^2)$.

2. Evaluation of Conjunctive Queries with Unary Relation Symbols

1. Let $Q() :- r_1^1(x_1), \ldots, r_1^{k_1}(x_1),\ \ldots,\ r_n^1(x_n), \ldots, r_n^{k_n}(x_n),\ s_1(a_1), \ldots, s_m(a_m)$ be a Boolean query with unary relation symbols, where $x_1, \ldots, x_n$ are variables and $a_1, \ldots, a_m$ are constants.

   The polynomial time algorithm for checking whether $\mathbf{I} \models Q$:

   ```
   for $i = 1$ to $m$ do
       if $a_i \notin s_i^{\mathbf{I}}$, return false
   for $i = 1$ to $n$ do
       evaluate Boolean query $Q'() :- r_i^1(x_i), \ldots, r_i^{k_i}(x_i)$, i.e.,
           $A = (r_i^1)^{\mathbf{I}} \cap \cdots \cap (r_i^{k_i})^{\mathbf{I}}$
       if $Q'$evaluates to false, i.e., $A = \emptyset$, return false
   return true
   ```

2. Evaluation of unary conjunctive queries with built-in predicates is NP-hard.

   We show it by reduction from the problem of 3-colorability of undirected graphs. Let $G = (V, E)$ be an undirected graph. We construct a database instance $\mathbf{I}$ and a Boolean query Q such that

   $$G \text{ is 3-colorable iff } \mathbf{I} \models Q$$

   First, we set $\mathbf{I} = \{\mathit{color}(r), \mathit{color}(g), \mathit{color}(b)\}$. Second, assume $V = \{v_1, \ldots, v_n\}$. For each vertex v_i, $1 \leq i \leq n$, we introduce a variable x_i. So, we set $Q() :- L, M$, where

   $$\begin{aligned} L &= \{\mathit{color}(x_1), \ldots, \mathit{color}(x_n)\} \\ M &= \{x_i \neq x_j \mid (v_i, v_j) \in E\} \end{aligned}$$

   Now, we prove that G is 3-colorable iff $\mathbf{I} \models Q$.

   ($\Rightarrow$) Assume G is 3-colorable. It follows that there exists an assignment $c\colon V \to \{r, g, b\}$ such that whenever $(v_i, v_j) \in E$ it holds that $c(v_i) \neq c(v_j)$. We can use this assignment to satisfy Q. More precisely, let $\alpha(x_i) = c(v_i)$, then it is easy to see that $\alpha(L) \subseteq \mathbf{I}$ and $\alpha \models M$.

   ($\Leftarrow$) Assume Q evaluates to true over $\mathbf{I}$. We can invert the argument above to show that G is 3-colorable.