Assignemnt 1
Due November 15, 2018

Question 1: Conjunctive Queries

Consider the following social-network signature:

\[
\text{Users}(\text{uid}) , \text{Follows}(\text{follower}, \text{followed}) , \text{Post}(\text{uid}, \text{pid}, \text{content}) , \text{Likes}(\text{uid}, \text{pid})
\]

We would like to find all the noticed users, where a user is noticed if she has at least one follower, or she posted a message that at least one user likes.

(a) Phrase a UCQ that finds the noticed users.
(b) Prove that this query cannot be phrased as a CQ (without the union).

Question 2: Data/Combined Complexity

Explain the source of confusion in the apparent contradiction entailed by the following correct statements.

1. Every query in Relational Algebra (RA) can be evaluated in polynomial time.
2. For every natural number \(k\), we can phrase in RA (over the edge relation \(E/2\)) the existence of a clique of size \(k\).
3. The maximum-clique problem is NP-complete, and hence, not solvable in polynomial time under conventional complexity assumptions

Question 3: Query Output Size

Let \(S\) be the relational schema that consists of three relations:

\[
R(A,B) , \ S(B,C) , \ T(C,A)
\]

(a) Show a construction of a database \(D(N)\) over \(S\) such that, given a natural number \(N\), both of the following hold.

- Each of the relations \(R^{D(N)}, S^{D(N)}\) and \(T^{D(N)}\) contains precisely \(N\) (distinct) tuples.
- The natural join of every two relations (i.e., \(R \Join S, S \Join T,\) and \(T \Join R\)) over \(D\) contains \(\Omega(N^2)\) tuples.

(b) Show that the result of the three-way join \(R \Join S \Join T\) over \(D\) contains \(O(N^{1.5})\) tuples, where \(N\) is the maximal number of tuples in a relation (i.e., \(R^D, S^D\) or \(T^D\)). (Hint: pick a column and split the analysis into two parts: values that repeat at least \(\sqrt{N}\) times, and the rest.)
(c) Show a construction of a database $D(N)$ over $S$ such that, given a natural number $N$, both of the following hold.

- Each of the relations $R^{D(N)}$, $S^{D(N)}$ and $T^{D(N)}$ contains precisely $N$ (distinct) tuples.
- The result of $R \bowtie S \bowtie T$ over $D$ contains $\Omega(N^{1.5})$ tuples.

(d) Show an algorithm for computing $R \bowtie S \bowtie T$ over a given $D$ in $o(N^2)$, that is, asymptotically faster than the algorithm that materializes a binary join.

(e) Suppose that $S$ had the functional dependency $R : A \rightarrow B$. Would it have an impact on the asymptotic size of the three-way join? Please explain your answer.

**Question 4: Combined and Parameterized Complexity**

The *Longest Common Subsequence Problem* (LCS) is the decision problem defined as follows. We are given as input:

- an alphabet $\Sigma = \{\sigma_1, \ldots, \sigma_m\}$ of characters;
- a sequence $s_1, \ldots, s_n$ of strings over (i.e., sequences of elements from) $\Sigma$;
- a natural number $k$.

A *common subsequence* is a sequence of characters that appear left-to-right, not necessarily in a contiguous block, in all of the $n$ strings. LCS is the problem of determining whether there is a common subsequence of length $k$.

For example, suppose that:

- $\Sigma = \{A, B, C, D\}$;
- $s_1 = ABAC$, $s_2 = CAABBA$, $s_3 = DACBCA$;
- $k = 3$.

Then the answer to this instance of LCS is “yes” due to the common subsequence $ABA$. LCS is known to be NP-complete, even on a binary alphabet [1].

**Question 3.a.** In this question you are required to reconstruct an argument that we developed in class. Suppose that there exists an algorithm $A$ for enumerating all common subsequences of a given length in *polynomial total time*. Prove that in this case, LCS can be solved in polynomial time (hence, the existence of $A$ is unlikely).

**Question 3.b.** Show that for every *fixed* alphabet $\Sigma$ (e.g., $\Sigma = \{0, 1\}$ or $\Sigma = \{A, G, C, T\}$) there is an FPT algorithm for solving LCS when the parameter is $k$.

Good luck!

**References**