Bazy danych 2024

Task List No. 3

Conjunctive queries are RRD queries built from atomic formulas (e.g., $R(x, y)$) and conjunctions and existential quantifiers. In general, atomic formulas can also include equalities and inequalities between constants and variables (e.g., $x = 5$, $x \neq y$, $x < z$) but on this list, we only allow relational atoms and do not allow the use of constants.

Consider a database representing a certain directed graph with edges stored in the relation $E(S, T)$. Unfortunately, in our queries, we cannot use the relation E. Instead, we have access to relations $P_i(x, y)$ for some $i > 1$. The relation $P_i(x, y)$ contains pairs of vertices connected by a path of length i, e.g., $P_2(x, z)$ could be defined as $(\exists y)E(x, y) \wedge E(y, z)$. This corresponds to a situation where, for example, due to security reasons, we only have access to the database through a set of perspectives (views), and access to the original relations is blocked.

If we want to compute answers to some query ψ using the relation E, we can try to modify (rewrite) ψ to use the symbols of available perspectives instead of E. For example, if we only have access to the perspective $P_2(x, y)$ and we want to express the query $P_4(x, z) = (\exists y_1, y_2, y_3)E(x, y_1) \wedge E(y_1, y_2) \wedge E(y_2, y_3) \wedge E(y_3, z)$, we can do it like this: $P_4'(x, z) = (\exists y)P_2(x, y) \wedge P_2(y, z)$ (notice that P_4' is also a conjunctive query and is equivalent to $P_4(x, y)$).

As a warm-up, show how to rewrite the query $P_7(x, y)$ using only perspectives $P_2(x, y)$ and $P_3(x, y)$.

1. (1 pt.) Show that there is no conjunctive query that uses only perspectives $P_3(x, y)$ and $P_4(x, y)$ as atomic formulas and which is equivalent to the query $P_5(x, y)$.

2. (1 pt.) Write an RRD query (allowed $\exists, \forall$, and all Boolean connectives) that uses only perspectives $P_3(x, y)$ and $P_4(x, y)$ and is equivalent to the query $P_5(x, y)$.

The domain of the database A, $\mathrm{dom}(A)$, is defined as the set of all elements contained in the tuples in the relations of A. A homomorphism between databases A and B is a function $h : \mathrm{dom}(A) \to \mathrm{dom}(B)$ satisfying the condition: for every relation symbol R and every tuple $(a_1, \ldots, a_n) \in \mathrm{dom}(A)^n$ (where n is the number of attributes of R) we have

$$A \models R(a_1, \ldots, a_n) \implies B \models R(h(a_1), \ldots, h(a_n))$$

3. (1 pt) During the lecture, the proof of the following implication was presented.

Let Q_1 and Q_2 be Boolean conjunctive queries (i.e., without free variables). If there exists a homomorphism from the canonical database C_{Q_2} to C_{Q_1}, then $Q_1 \subseteq Q_2$.

Prove the implication in the reverse direction. You can use the fact that for any Boolean conjunctive query Q and database D we have $D \models Q$ (i.e., the query is satisfied in D) if and only if there exists a homomorphism from the canonical database C_Q to the database D.

4. (1 pt)
1. Show that there exists a graph G such that for any graph G' there exists a homomorphism from G' to G.
2. Prove or provide a counterexample to the statement: for any graphs G_1 and G_2, the following conditions are equivalent:
   - there exist homomorphisms $h_1 : G_1 \to G_2$ and $h_2 : G_2 \to G_1$ (such graphs are called *homomorphically equivalent*)
   - there exists an isomorphism $f : G_1 \to G_2$.

5. (1 pt) In this task, we consider symmetric graphs, i.e., those for which if there is an edge from v to w then there is also an edge from w to v. Show that any two such graphs that are cycles of even length are homomorphically equivalent.

6. (1 pt) A propositional calculus formula is in 3CNF form when it is in conjunctive normal form, and each clause additionally contains at most 3 variables. An example of a 3CNF formula: $(p \vee \neg q \vee r) \wedge (s \vee \neg r)$.

Consider the Boolean satisfiability problem known as 3SAT: given a formula in 3CNF form, decide whether there exists an assignment satisfying it.

Show that the 3SAT problem can be expressed as the problem of the existence of a homomorphism between databases. In other words, demonstrate that if you can solve the problem of the existence of a homomorphism, then you can also solve the 3SAT problem.

Your construction should work in polynomial time.

7. (0.5 points) It is known that the 3SAT problem is computationally hard. Clay Institute has offered a prize of one million dollars for demonstrating the existence of any polynomial-time algorithm solving this problem. On the other hand, the problem of evaluating conjunctive queries (i.e., the existence of a homomorphism) is not considered particularly difficult, and database engines have been efficiently computing answers to SQL queries for many years. However, in task 6, we essentially showed that the 3SAT problem is easier (or equally difficult) as the problem of the existence of a homomorphism. Explain why this is not a paradox (and solving task 6 alone is not sufficient to win the prize).