Database Theory: Exercise Sheet 1

Anuj Dawar
anuj.dawar@cl.cam.ac.uk

February 2, 2004

1. (The exercise on Slide 34). Suppose ϕ is a conjunctive formula, x is a free variable occurring in ϕ, $\mathcal{I}$ a database instance and ν a valuation of the variables such that $\nu(x) \notin adom(\mathcal{I})$. Prove, by induction on the structure of ϕ, that $\mathcal{I} \not\models_\nu \phi$.

2. (The exercise on Slide 64). Let P be a Datalog program, and T_P be defined as on slide 63. If $\mathcal{I}$ and $\mathcal{J}$ are two database instances such that $\mathcal{I} \subseteq \mathcal{J}$ and R is a relation in $idb(P)$, show that $T_P(\mathcal{I})(R) \subseteq T_P(\mathcal{J})(R)$.

3. Prove the assertion on slide 72. That is, if $\mathcal{I}$ and $\mathcal{J}$ are such that $\mathcal{I} \subseteq \mathcal{J}$ and they agree on all *extensional* relations and P is a semipositive Datalog program, then for each R in $idb(P)$ $T_P(\mathcal{I})(R) \subseteq T_P(\mathcal{J})(R)$. Use this to show that any query defined by a semipositive Datalog program is monotone in the restricted sense of slide 73.

4. Consider the following questions one might ask of the *Cinema* database:

   - Is there a film directed by Almodovar playing in Cambridge?
   - Which directors have appeared in every film they've directed?
   - List all films in which Allen acted or directed.

   (a) For each of these, write a relational calculus query that defines it.

   (b) For each of the queries you wrote down, state whether or not it is safe, whether or not it is domain-independent and whether or not it is generic.

   (c) Which of these queries is definable in the conjunctive calculus? Which of them is definable in Datalog? For each such, write a Datalog program to define it.

   (d) For any query in the list that is not definable in Datalog, prove that it is not (by constructing an appropriate example).

   (e) For queries above not definable in Datalog, state whether they are definable in semipositive Datalog. Either give a program or a reason why it's inexpressible.

5. Given the *Railway* database of slide 56, write Datalog programs to compute the following queries:

(a) List all stations that are reachable from both Oxford and Cambridge.

(b) List all stations that are reachable from either Oxford or Cambridge.

6. Suppose we have a database with two relations. One is the *Railway* relation of slide 56. The other is a similar relation *Coach*[*Service,From,To*]. Write programs in stratified Datalog to express the following queries:

   (a) List the pairs of stations x, y such that one can go from x to y by train but not by coach.

   (b) List the pairs of stations x, y such that one can go from x to y by coach in such a way that for no leg of the journey is there a possible train connection.

7. For the queries in 5 above, write expressions of the fixed-point calculus.

8. For all the queries in 5 and 6 above, give an upper bound on the complexity of evaluating the query.

9. You are to prove that Duplicator has a winning strategy in the game described on slide 97. Let $\text{dist}(u, v)$ denote the distance from u to v in a graph (which is ∞ if there is no path from u to v). Show, by induction on r, that, if $u_1, \ldots, u_r$ are nodes in G_m and $v_1, \ldots, v_m$ are nodes in H_m that have been selected in the first r rounds of the Ehrenfeucht game, then Duplicator can inductively maintain the condition that for all i: either $\text{dist}(u_i, u_{i+1}) = \text{dist}(v_i, v_{i+1})$ or $\text{dist}(u_i, u_{i+1}), \text{dist}(v_i, v_{i+1}) \geq 2^{m-r}$. Explain why this implies that for $r \leq m$ rounds of the game, the pair (u_i, u_j) is an edge in the graph G_m if, and only if, (v_i, v_j) is an edge in H_m.

10. Consider the query on top of slide 86 (*Is there a way to travel around Britain so that I visit every railway station exactly once*?). Is this query monotone in the sense of slide 68? Is it monotone in the restricted sense of slide 72? What can you conclude about it's definability in Datalog and semipositive Datalog?

11. Take it as given (as stated on slide 97) that we can construct, for each m two graphs G_m and H_m, one consisting of a single cycle and one consisting of two disjoint cycles such that **Duplicator** has a winning strategy in the m-move Ehrenfeucht game played on these two graphs. Show why this implies that the query at the top of slide 86 (*Is there a way to travel around Britain so that I visit every railway station exactly once*?) is not definable in the relational calculus.