Exercices

Databases 2 tutorial, M2 Data Science

Mikaël Monet

Notations. We usually use letters from the beginning of the alphabet $(a, b, c, d, \ldots)$ to denote constants and from the end $(t, u, v, x, y, z, \ldots)$ to denote variables.

Homomorphisms for conjunctive queries with free variables. Recall that a conjunctive query (CQ) is a first-order query of the form $q(\bar{x}) := \exists \bar{y} \bigwedge_{i=1}^{m} R_i(\bar{z_i})$, where $\bar{x}$ is a tuple that contains all the free variables of $\exists \bar{y} \bigwedge_{i=1}^{m} R_i(\bar{z_i})$. For instance, $q(x, y, x) := \exists z, t : \ R(x, z, t) \wedge R(y, a, t) \wedge S(x, z)$ is one such query.

Q1. What is $q(D)$, for D the following database?

R		
a	a	c
b	c	c
c	c	c

S	
a	a
b	c
b	b
c	a

Q2. How would you write the query $q(x, y, x)$ above in SQL?

Q3. Propose a notion of homomorphism between a CQ $q(\bar{x})$ and pair $(D, \bar{a})$ consisting of a database D and a tuple of constants $\bar{a}$, with $|\bar{a}| = |\bar{x}|$, such that we have $\bar{a} \in q(D)$ if and only if such a homomorphism exists. You can use the notation $q(\bar{x}) \overset{h}{\hookrightarrow} (D, \bar{a})$ to denote the existence of such a homomorphism h.

Given two CQs $q_1(\bar{x_1}), q_2(\bar{x_2})$ with $|\bar{x_1}| = |\bar{x_2}|$, recall that $q_1(\bar{x_1})$ is *contained in* $q_1(\bar{x_1})$ (written $q_1(\bar{x_1}) \subseteq q_2(\bar{x_2})$) if, for every database D, we have $q_1(D) \subseteq q_2(D)$.

Q4 [Homomorphism theorem for non-Boolean CQs]. Propose a notion of homomorphism between CQs, written $q_1(\bar{x_1}) \overset{h}{\hookrightarrow} q_2(\bar{x_2})$, such that we have $q_2(\bar{x_2}) \subseteq q_1(\bar{x_1})$ iff there exists h such that $q_1(\bar{x_1}) \overset{h}{\hookrightarrow} q_2(\bar{x_2})$ (and prove it).

Q5. Explain why the containment problem for CQs (not necessarily Boolean) is NP-complete. (You can use the fact that Containment for Boolean CQs is NP-complete).

Conjunctive queries with disequalities. A *Boolean CQ with disequalities*, written BCQ$^{\neq}$, is a Boolean conjunctive query (BCQ) in which we can additionaly impose that some variables should be mapped to distinct constants. For instance $q_4 := \exists x, y R(y, x, c) \wedge R(x, x, x) \wedge x \neq y$.

Q6. Do we have $D \models q_4$ for the above database?

Q7. Propose a notion of homomorphism between a BCQ$^{\neq}$ q and a database D such that we have $D \models q$ if and only if such a homomorphism exists.

We now propose the following definition of a homomorphism from a BCQ$^{\neq}$ q_1 to another BCQ$^{\neq}$ q_2: it is a homomorphism (in the sense of **Q7**) between q_1 and the canonical database of q_2.

Q8. Is the analogue of the homomorphism theorem for this notion of homomorphism and BCQ$^{\neq}$s true? (If yes prove it, if not, provide a counterexample.)

Unions of conjunctive queries. A Boolean Union of Conjunctive Queries (UCQ) is a first-order query of the form $q := \bigvee_{i=1}^{m} q_i$, where each q_i is a BCQ. For instance, $q := [\exists x, y R(x, y, c) \wedge S(y, x)] \vee [\exists t S(t, t)]$ is such a query. These correspond to the SQL queries formed with keywords SELECT, FROM, WHERE, UNION, where we only use equality in the WHERE clause.

Q9. Prove that ModelChecking(UCQs) (i.e., in combined complexity) is NP-complete.

Q10. Consider two UCQs $q = \bigvee_{i=1}^{n} q_i$ and $q' = \bigvee_{i=1}^{n'} q'_i$. Prove that we have $q \subseteq q'$ iff for every $i \in \{1, \ldots, n\}$, there exists $j \in \{1, \ldots, n'\}$ such that $q_i \subseteq q'_j$.

Q11. Prove that Containment(UCQs) (i.e., in combined complexity) is NP-complete (hint: use **Q10.**).

Cores. **Q12.** For each of the following BCQs, compute a core:

1. $q_1 = \exists x, y, z R(z, y, y) \wedge R(x, y, z)$
2. $q_2 = \exists x_1, x_2, y_1, y_2, z_1, w_1 : \; E(x_1, y_1) \wedge E(y_1, z_1) \wedge E(z_1, w_1) \wedge E(w_1, x_1) \wedge E(x_2, y_2) \wedge E(y_2, x_2)$ (hint: draw the query as a graph to better see it.)

We saw in the course that containment of BCQs is NP-complete, and to compute a core of a BCQ we need to solve multiple times the containment problem for BCQs. This does not directly show that computing a core is NP-hard, because the instances of Containment that we have to solve are of a very restricted shape: we always want to determine whether $q' \subseteq q$ for queries such that $A_{q'} \subseteq A_q$ (recall that A_q denotes the set of atoms of q). It turns out that this restriction does not make the problem easier:

Q13. Prove that the following problem is NP-complete.
INPUT: Two BCQs q_1, q_2 such that $A_{q_1} \subseteq A_{q_2}$.
OUTPUT: YES if $q_1 \subseteq q_2$, NO otherwise.