Homework 2

Tien Thuy Ho

May 18, 2018

Problem 1

1. All drinkers who frequent at least one bar that serves at least one beer they like.

$$\{x : \exists y\, \texttt{FREQUENTS}(x, y) \wedge \exists z\, (\texttt{LIKES}(x, z) \wedge \texttt{SERVES}(y, z))\}$$

   This is a conjunctive query because it is expressed in $\wedge$ and $\exists$ only.

2. All drinkers who frequent only bars that serve at least one beer they like.

$$\{x : \forall y\, \texttt{FREQUENTS}(x, y) \rightarrow \exists z\, (\texttt{SERVES}(y, z) \wedge \texttt{LIKES}(x, z))\}$$

   This is not a conjunctive query because it includes the universal quantifier $\forall$ and implication $\rightarrow$.

3. All drinkers who frequent every bar that serves ANCHOR STEAM.

$$\{x : \forall y\, (\texttt{SERVES}(y, \texttt{ANCHOR STEAM})) \wedge \texttt{FREQUENTS}(x, y))\}$$

   This is not a conjunctive query because it includes the universal quantifier $\forall$.

Problem 2

Let R be a relational schema with attributes A, B, and C

$$\pi_A(\pi_{A,C}(R) \bowtie \pi_{B,C}(R))$$

1. Conjunctive query as a relational calculus:

$$\{x : \exists y \exists z\, (R(x, y) \wedge R(z, y))\}$$

2. Conjunctive query as rule:

$$q(x) : -R(x, y), R(z, y)$$

Problem 3

Acknowledgment: http://www.cs.rhul.ac.uk/home/green/mathscsp/slides/tutorials/kolaitis/csp-oxford.pdf

1. Let φ be a 2-CNF formula that involves variables $x_1, ..., x_n$. Using the relational database schema S with three binary relational schemas, we can rewrite φ as a database instance of S:

$$S^\varphi = \{A_1, A_2, A_3\}$$

where

$$A_1 = \{(x, y) : (x \vee y) \in \varphi\}$$
$$A_2 = \{(x, y) : (\neg x \vee y) \in \varphi\}$$
$$A_3 = \{(x, y) : (\neg x \vee \neg y) \in \varphi\}$$

Note that each relation A_i represents a possible conjunct found in a 2-CNF formula. For each of these conjuncts, we create a relation that contains all satisfiable combinations of boolean values as tuples. The result is another database instance of S:

$$S^* = \{B_1, B_2, B_3\}$$

where

$$B_1 = \{(\texttt{true}, \texttt{false}), (\texttt{true}, \texttt{true}), (\texttt{false}, \texttt{true})\}$$
$$B_2 = \{(\texttt{false}, \texttt{true}), (\texttt{true}, \texttt{true}), (\texttt{false}, \texttt{false})\}$$
$$B_3 = \{(\texttt{false}, \texttt{true}), (\texttt{true}, \texttt{false}), (\texttt{false}, \texttt{false})\}$$

It follows that φ is satisfiable if there is a homomorphism $h : S^\varphi \to S^*$ such that for every $i \leq 3$:

$$(x_1, x_2) \in A_i \Rightarrow (h(x_1), h(x_2)) \in B_i$$

Let's consider an example of $\varphi = (x \vee y) \wedge (x \vee \neg y)$. We have

$$S^\varphi = \{A_1, A_2, A_3\}$$

where

$$A_1 = \{(x, y)\}$$
$$A_2 = \{(y, x)\}$$
$$A_3 = \{\}$$

There is a homomorphism $h : S^\varphi \to S^*$ where $h(x) = \texttt{true}$ and $h(y) = \texttt{false}$. Hence φ is satisfiable.

2. Similarly, 3-SAT can be viewed as a special case of the HOMOMORPHISM PROBLEM in the same way that we have demonstrated for 2-SAT. One change would be that the relational database schema S would have four trinary relational schemas. The database instance of S to represent a 3-CNF formula φ would be

$$S^\varphi = \{A_1, A_2, A_3, A_4\}$$

where A_i represents a possible conjunct found in a 3-CNF formula

$$A_1 = \{(x, y, z) : (x \vee y \vee z) \in \varphi\}$$
$$A_2 = \{(x, y, z) : (\neg x \vee y \vee z) \in \varphi\}$$
$$A_3 = \{(x, y, z) : (\neg x \vee \neg y \vee z) \in \varphi\}$$
$$A_4 = \{(x, y, z) : (\neg x \vee \neg y \vee \neg z) \in \varphi\}$$

The database instance of S to represent the constraints for each conjunct is defined as following:

$$S^* = \{B_1, B_2, B_3, B_4\}$$

where

$$B_1 = \{\texttt{true}, \texttt{false}\}^3 - (\texttt{false}, \texttt{false}, \texttt{false})$$
$$B_2 = \{\texttt{true}, \texttt{false}\}^3 - (\texttt{true}, \texttt{false}, \texttt{false})$$
$$B_3 = \{\texttt{true}, \texttt{false}\}^3 - (\texttt{true}, \texttt{true}, \texttt{false})$$
$$B_4 = \{\texttt{true}, \texttt{false}\}^3 - (\texttt{true}, \texttt{true}, \texttt{true})$$

Problem 4

NATURAL JOIN NON-EMPTINESS: Given relations $R_1, ..., R_m$, is their natural join $R_1 \bowtie ... \bowtie R_m$ non-empty?

1. NATURAL JOIN NON-EMPTINESS is in NP.

   Proof: We can guess a tuple $(a_1, ..., a_k)$, where k is the number of all attributes, and check the membership of each value in the given relations. For each value a_i of attribute B, we iterate through all the relations. If we encounter a relation R_j that has the same attribute as a_i, then we check if $a_i \in \pi_B(R_j)$. We check a_i against all the relations that contain attribute B. The tuple $(a_1, ..., a_k) \in R_1 \bowtie ... \bowtie R_m$ if the membership test for each value in the tuple is positive. It then follows that $R_1 \bowtie ... \bowtie R_m$ is non-empty. This checking method should run in polynomial time $O(km)$, where k is the size of the tuple and m is the number of relations.

2. NATURAL JOIN NON-EMPTINESS is an NP-hard problem.

   Proof: We need to show that NATURAL JOIN NON-EMPTINESS can be reduced to one of the NP-complete problem in polynomial time. Let's use Integer Linear Inequalities (ILE) as the NP-complete problem for reduction. We need to show that ILE $\preceq_p$ NATURAL JOIN NON-EMPTINESS. Let φ be a system of m linear inequalities $l_1, ..., l_m$ where in each inequality l_i for $1 \leq i \leq m$, there are an arbitrary number k of variables x_{i_j} for $1 \leq j \leq k$. We can transform each inequality l_i to a corresponding relation R_i with the arity equal to the number of variables in l_i. Each tuple in R_i is a solution to the inequality. For example, if l_i is $x_1 + x_2 > 0$, then we have the corresponding R_i with two attributes x_1 and x_2 and all tuples (x_1, x_2) that satisfy this inequality. Let φ^* be the resulting natural join of m relations. It follows that φ^* is a set of tuples that satisfy all the inequalities.

   $$\varphi \text{ has a solution} \Longleftrightarrow \varphi^* \text{ is non-empty}$$

   Hence, ILE $\preceq_p$ NATURAL JOIN NON-EMPTINESS.

3. NATURAL JOIN NON-EMPTINESS is NP-hard even if the given relations are of arity at most 3.

   Proof: Since we have showed that the general NATURAL JOIN NON-EMPTINESS is a NP-hard problem, we need to show that NATURAL JOIN NON-EMPTINESS $\preceq_p$ 3-NATURAL JOIN NON-EMPTINESS. Let φ be the natural join $R_1 \bowtie ... \bowtie R_m$ of relations $R_1, ..., R_m$. If a relation has an arity of more than 3, we can break it up into multiple relations of arity of 3. For instance, given a relation $R_i(A, B, C, D, E)$ where A, B, C, D and E are the attributes, we replace it with $R_{i_1}(A, B, C)$ and $R_{i_2}(C, D, E)$ such that the two relations share the same attribute and all its values. In that way, the natural join $R_{i_1} \bowtie R_{i_2}$ gives back $R_i(A, B, C, D, E)$. Let φ^* be the resulting natural join of relations of arity at most 3, then

$$\varphi \text{ is non-empty} \iff \varphi^* \text{ is non-empty}$$

Hence, NATURAL JOIN NON-EMPTINESS $\preceq_p$ 3-NATURAL JOIN NON-EMPTINESS.

4. Let's consider the problem of NATURAL JOIN NON-EMPTINESS in which we are given at most M relations and each relation has arity at most K. The naive algorithm of computing the natural join of relations is to compare every tuple of every relation against one another and check if they agree on common attributes. Let I be a database instance from which relations $R_1, ..., R_M$ are derived. Then the number of possible tuples for each relation of arity at most K is $|adom(I)|^K$. Since we take the natural join of at most M relations, the total number of tuple comparisons is $(|adom(I)|^K)^M$, or $|adom(I)|^{KM}$. Hence, even in the worst case of exhausting all the possible tuples that each relation can have, we can compute the natural join of at most M relations, each with arity at most K, in polynomial time with respect to the size of the active domain of a database.

Problem 5

Let q_1 and q_2 be the Boolean conjunctive queries:

$$q_1 : -E(x_1, x_2), E(x_2, x_1), E(x_2, x_3), E(x_3, x_2), E(x_3, x_4), E(x_4, x_3), E(x_4, x_1), E(x_1, x_4).$$

and

$$\begin{aligned} q_2 : -&E(x_1, x_2), E(x_2, x_1), E(x_2, x_3), E(x_3, x_2), E(x_3, x_4), \\ &E(x_4, x_3), E(x_4, x_1), E(x_1, x_4), E(x_1, x_3), E(x_3, x_1). \end{aligned}$$

Proof: The canonical instances of q_1 and q_2 are given as

$$\begin{aligned} I^{q_1} =& E(x_1, x_2), E(x_2, x_1), E(x_2, x_3), E(x_3, x_2), \\ & E(x_3, x_4), E(x_4, x_3), E(x_4, x_1), E(x_1, x_4) \\ I^{q_2} =& E(x_1, x_2), E(x_2, x_1), E(x_2, x_3), E(x_3, x_2), E(x_3, x_4), \\ & E(x_4, x_3), E(x_4, x_1), E(x_1, x_4), E(x_1, x_3), E(x_3, x_1) \end{aligned}$$

1. $q_1 \not\subseteq q_2$
   If we use a function h where

   $h(x_1) = x_1$
   $h(x_2) = x_2$
   $h(x_3) = x_3$
   $h(x_4) = x_4$

   to map most tuples from I^{q_2} to I^{q_1}, then we cannot map (x_1, x_3) and (x_3, x_1) from I^{q_2} to I^{q_1}. Hence, there does not exist any homomorphism h: $I^{q_2} \to I^{q_1}$. Hence it follows that $q_1 \not\subseteq q_2$.

2. $q_2 \subseteq q_1$
   There is homomorphism h: $I^{q_1} \rightarrow I^{q_2}$ where

   $h(x_1) = x_1$
   $h(x_2) = x_2$
   $h(x_3) = x_3$
   $h(x_4) = x_4$

   Hence, $q_2 \subseteq q_1$ by the Homomorphism Theorem.

3. A conjunctive query q_3 such that q_3 is a minimal conjunctive query equivalent to q_1

$$q_3 : -E(x_1, x_2), E(x_2, x_1).$$

   (a) We need to show that $q_1 \equiv q_3$. The canonical instance of q_3 is $I^{q_3} = E(x_1, x_2), E(x_2, x_1)$. There is a homomorphism $h_1 : I^{q_1} \rightarrow I^{q_3}$ where

   $h_1(x_1) = x_1$
   $h_1(x_2) = x_2$
   $h_1(x_3) = x_1$
   $h_1(x_4) = x_2$

   Hence, $q_3 \subseteq q_1$ by the Homomorphism Theorem (1). There is also a homomorphism h_2: $I^{q_3} \rightarrow I^{q_1}$ where

   $h_2(x_1) = x_1$
   $h_2(x_2) = x_2$

   Therefore, $q_1 \subseteq q_3$ by the Homomorphism Theorem (2). By (1) and (2), $q_1 \equiv q_3$.

   (b) To show that q_3 is minimal, we consider all other conjunctive queries that are equivalent to q_1

   $q_3' : -E(x_1, x_2), E(x_2, x_1), E(x_2, x_3), E(x_3, x_2).$
   $q_3'' : -E(x_1, x_2), E(x_2, x_1), E(x_2, x_3), E(x_3, x_2), E(x_3, x_4), E(x_4, x_3).$

   q_3 has the fewest conjuncts compared to all the conjunctive queries equivalent to q_1.

4. A conjunctive query q_4 such that q_4 is a minimal conjunctive query equivalent to q_2

$$q_4 : -E(x_1, x_2), E(x_2, x_1), E(x_2, x_3), E(x_3, x_2), E(x_1, x_3), E(x_3, x_1).$$

   (a) We need to show that $q_2 \equiv q_4$. The canonical instance of q_4 is $I^{q_4} = E(x_1, x_2), E(x_2, x_1)$, $E(x_2, x_3), E(x_3, x_2), E(x_1, x_3), E(x_3, x_1)$. There is a homomorphism $h_1 : I^{q_2} \rightarrow I^{q_4}$ where

   $h_1(x_1) = x_1$
   $h_1(x_2) = x_2$
   $h_1(x_3) = x_3$
   $h_1(x_4) = x_2$

By the Homomorphism Theorem, $q_4 \subseteq q_2$. There is also a homomorphism h_2: $I^{q_4} \to I^{q_2}$ where

$$h_2(x_1) = x_1$$
$$h_2(x_2) = x_2$$
$$h_2(x_3) = x_3$$

Hence, $q_2 \subseteq q_4$ by the Homomorphism Theorem.

(b) To show that q_4 is minimal, we consider all other conjunctive queries that are equivalent to q_2

$$q_4' : -E(x_1, x_2), E(x_2, x_1), E(x_2, x_3), E(x_3, x_2), E(x_3, x_4), E(x_4, x_3), E(x_1, x_3), E(x_3, x_1).$$

q_4 has the fewest conjuncts compared to all the conjunctive queries equivalent to q_2.

Problem 6

Given a Boolean conjunctive query q

$$q : -q_1, q_2, ..., q_n.$$

1. If both q_1 and q_2 are minimal equivalent conjunctive queries of q, then the canonical database of q_1 is isomorphic to the canonical database of q_2.

   Proof: Let q_1 and q_2 be the two minimal equivalent conjunctive queries to q. It follows that $q_1 \equiv q_2$. Hence, we have $q_1 \subseteq q_2$ and $q_2 \subseteq q_1$. By the Homomorphism Theorem, there is a homomorphism h_1: $I^{q_1} \to I^{q_2}$, and there is also a homomorphism h_2: $I^{q_2} \to I^{q_1}$. We have
   $$h_1 \circ h_2 = I^{q_2} \to I^{q_2} = id_{I^{q_2}}$$
   Hence, h_2 is the inverse of h_1, and h_1 is bijective. Since there exists an isomorphism h_1: $I^{q_1} \to I^{q_2}$, q_1 and q_2 are isomorphic.

2. There is a Boolean conjunctive query q' that is a minimal equivalent query to q and is obtained from q by removing zero or more conjuncts of q.

   Proof: Any conjunctive query has an equivalent query of the same number of atoms by renaming its variables. Therefore, the size of a minimal equivalent conjunctive query should be at most the size of q. We can obtain a minimal equivalent conjunctive query in a greedy way. We iterate through all atoms of q and see if we can obtain another equivalent query by removing one atom. If another equivalent query is found, we repeat the same process for the remainting atoms. Otherwise, we have found the minimal one.

Problem 7

Let q be a union $q_1 \cup ... \cup q_m$ of conjunctive queries $q_1, ..., q_m$. This union is non-redundant if there is no pair (i, j) such that $i \neq j$ and $q_i \subseteq q_j$,

1. The union

$$E(x, y) \vee \exists z\ (E(x, z) \wedge E(z, y)) \vee \exists z, w\ (E(x, z) \wedge E(z, w) \wedge E(w, y))$$

is redundant.

Proof: Let's consider the two conjunctive queries in the union

$$q_1 = E(x, y)$$
$$q_2 = \exists z\,(E(x, z) \wedge E(z, y))$$

There is a homomorphism h: $I^{q_1} \to I^{q_2}$ where

$h(x) = z$
$h(y) = y$

By the Homomorphism Theorem, $q_2 \subseteq q_1$. This violates the definition of a non-redundant union.

2. Assume that q is a non-redundant union $q_1 \cup ... \cup q_m$ of Boolean conjunctive queries $q_1, ..., q_m$ and that q' is a non-redundant union $q'_1 \cup ... \cup q'_n$ of Boolean conjunctive queries $q'_1, ..., q'_n$ such that q is equivalent to q'.

   (a) For each $i \leq m$, there is $j \leq n$ such that $q_i \equiv q'_j$.

   Proof: Since $q \equiv q'$, we have $q_1 \cup ... \cup q_m \equiv q'_1 \cup ... \cup q'_n$. It follows that $q_1 \cup ... \cup q_m \subseteq q'_1 \cup ... \cup q'_n$ and $q'_1 \cup ... \cup q'_n \subseteq q_1 \cup ... \cup q_m$. By the theorem of Sagiv and Yannakakis for unions of conjunctive queries, for each $i \leq m$, there is $j \leq n$ such that $q_i \subseteq q'_j$. Also, for each $j \leq n$, there is $i \leq m$ such that $q'_j \subseteq q_i$. It follows that for each $i \leq m$, there is $j \leq n$ such that $q_i \subseteq q'_j$, where $q'_j \subseteq q_k$ for some $k \leq m$. Towards contradiction, let's assume that $i \neq k$. Then we have $q_i \subseteq q'_j \subseteq q_k$, or $q_i \subseteq q_k$. This violates the fact that q is a non-redundant union. Therefore, i must be equal to k. Hence, $q_i \subseteq q'_j$, where $q'_j \subseteq q_i$. This establishes that for each $i \leq m$, there is $j \leq n$ such that $q_i \equiv q'_j$.

   (b) $m = n$

   Proof: We have showed that for each $i \leq m$, there is $j \leq n$ such that $q_i \equiv q'_j$. Towards contradiction, let's assume that there is a $j \leq n$ such that $q_a \equiv q'_j$ and $q_b \equiv q'_j$ for some $a, b \leq m$ and $a \neq b$. Then $q_a \equiv q_b$, and $q_a \subseteq q_b$. This violates the fact that q is a non-redundant union. Hence, no two conjunctive queries in q are equivalent to the same conjunctive query in q'. It follows that there is a one-to-one equivalence mapping between conjunctive queries in q and q'. Hence, q and q' have the same number of conjunctive queries, and $m = n$.

   (c) i. For the first fact, from the theorem of Sagiv and Yannakakis, we know that for each $j \leq n$, there is $i \leq m$ such that $q_i \subseteq q'_j$, where $q'_j \subseteq q_k$ for some $k \leq m$. Our goal is to show that $i = k$ to be able to prove $q_i \equiv q'_j$. However, without the assumption that q is a non-redundant union, q_i could be a subset of q_k where $i \neq k$. Hence, we cannot establish that $q'_j \subseteq q_i$.

   ii. For the second fact, without the assumption that q is a non-redundant union, two conjunctive queries of q could be equivalent to the same conjunctive query in q'. Thus q' can have fewer conjunctive queries and still guarantees for each $i \leq m$, there is $j \leq n$ such that $q_i \equiv q'_j$.