2. Positive and Conjunctive Queries

These are sample solutions to some of the exercises that were given as coursework. They are not intended as models but show each one way to approach the problem set in the exercise.

2. Unions of Conjunctive Queries

Show that adding union to simple conjunctive queries strictly increases the expressivity of the resulting query language. (Recall from the previous exercise that simple conjunctive queries have neither equality nor disequality atoms.)

Hint 1: Consider the query defined by the two rules

$$\begin{array}{lcl} Q(x) & :- & p(x) \\ Q(x) & :- & r(x) \end{array}$$

and show that no query defined by a single rule is equivalent to it.

Hint 2: Assume there is an equivalent simple conjunctive query. Then consider several databases distinguished by the atoms occurring in them.

Sample solution.

Let us recall several definitions. We view a database instance as a set of ground atoms and a disjunctive query as a set of conjunctive queries. We define the answer set of a disjunctive query as the union of the answer sets of all conjunctive queries in it.
The following proposition gives a solution for the exercise.

Proposition 1. *There is a disjunctive query that is not equivalent to any conjunctive query.*

Proof. Consider the query Q' given by the following two rules:

$$\begin{array}{lcl} Q'(x) & :- & p(x) \\ Q'(x) & :- & r(x). \end{array}$$

Assume there is a rule-based conjunctive query Q that is equivalent to Q'. The query Q is of the form

$$Q() :- R_1(\vec{x}_1), ..., R_n(\vec{x}_n). \tag{1}$$

Let us consider the two database instances $\mathbf{I}_1 = \{p(a)\}$ and $\mathbf{I}_2 = \{r(a)\}$. By our assumption $Q' \equiv Q$. Therefore, $Q'(\mathbf{I}_i) = Q(\mathbf{I}_i)$, for $i \in \{1, 2\}$.
For the first database instance, $Q'(\mathbf{I}_1) = \{\langle a \rangle\} = Q(\mathbf{I}_1)$. Therefore, Q's body can only contain atoms of the relation p. Similarly, for the second database instance we have $Q'(\mathbf{I}_2) = \{\langle a \rangle\} = Q(\mathbf{I}_2)$. Therefore, Q's body can only contain atoms of the relation r. Since p and r are distinct, we have obtained a contradiction. $\square$

3. Classes of Conjunctive Queries

We view queries as functions that map database instances to relation instances. Consider the following classes of conjunctive queries, which are distinguished by the form of the rules by which they can be defined:

CQ: rules without equality "$=$" and disequality "$\neq$" atoms ("simple" conjunctive queries);

CQ$_=$: rules that may have equality atoms, but no disequality atoms;

CQ$_{rep}$: rules that may repeat variables in the head, but do not have equality and disequality atoms;

CQ$_{const}$: rules that may have constants in the head, but do not have equality and disequality atoms;

CQ$_{rep,const}$: rules that may repeat variables and may have constants in the head, but do not have equality and disequality atoms.

Determine which inclusions hold between these classes and which not:

- To show that class C_1 is included in class C_2 (i.e., $C_1 \subseteq C_2$), indicate how any query in C_1 can be equivalently expressed by a query in C_2.
- To show that C_1 is not included in C_2 (i.e., $C_1 \not\subseteq C_2$), identify first a property P_2 such that all queries in C_2 have property P_2, and then exhibit a query in C_1 that does not property P_2.

Clearly, some inclusions are obvious. Note also that you can derive some other inclusions exploiting the fact that set inclusion is transitive.

Sample solution.

Claim. *The following inclusions hold:*

- $\mathsf{CQ} \subset \mathsf{CQ}_{rep}$, $\mathsf{CQ} \subset \mathsf{CQ}_{const}$,
- $\mathsf{CQ}_{rep} \subset \mathsf{CQ}_{rep,const}$, $\mathsf{CQ}_{const} \subset \mathsf{CQ}_{rep,const}$,
- $\mathsf{CQ}_{rep} \subset \mathsf{CQ}_{=}$.

Moreover, all inclusions hold that follow by transitivity from the inclusions above. However, no other inclusions hold. In particular, all the above inclusions are strict.

There is one inclusion that is not obvious is $\mathsf{CQ}_{rep} \subset \mathsf{CQ}_{=}$.

Lemma 1. *Let $Q(\bar{x}) \coloneq L, M$ be a conjunctive query with (possibly) equality atoms that has a repetition of the variable x_1 in the head. Then Q is equivalent to a query*

$$Q'(\bar{x}') \coloneq L, L', M, M', x_1 = x_1'$$

that has one repeated variable occurrence less than Q.

Proof. Without loss of generality assume that $\bar{x} = (x_1, x_1, x_3, \ldots, x_k)$, where some variables among $x_3, \ldots, x_k$ are possibly identical. Let x_1' be a fresh variable. Define the vector of variables $\bar{x}' = (x_1, x_1', x_3, \ldots, x_k)$. Moreover, define the set of atoms $L' := [x_1/x_1']L$ and the set of equalities $M' := [x_1/x_1']M$, that is, L', M' are obtained from L, M by replacing x_1 with the fresh variable x_1'. Moreover, define Q' as $Q'(\bar{x}') \coloneq L, L', M, M', x_1 = x_1'$. Clearly, Q' has one repetition in the head less than Q.
Now, let $\mathbf{I}$ be an instance, α be an assignment for the variables in Q, and suppose that $\mathbf{I}, \alpha \models L$ and $\alpha \models M$. We extend α to an assignment α' of the variables in Q' by also mapping x_1' to $\alpha(x_1)$. Then $\mathbf{I}, \alpha' \models L'$ and $\alpha' \models M', x_1 = x_1'$. This shows that $Q(\mathbf{I}) \subseteq Q'(\mathbf{I})$. Conversely, let α' be an assignment such that $\mathbf{I}, \alpha' \models L, L'$ and $\alpha' \models M, M', x_1 = x_1'$. Then $\mathbf{I}, \alpha' \models L$ and $\alpha' \models M$, hence $\alpha'(\bar{x}) \in Q(\mathbf{I})$. Since $\alpha'(x_1') = \alpha'(x_1)$, we have $\alpha'(\bar{x}') = \alpha'(\bar{x})$, and thus $Q'(\mathbf{I}) \subseteq Q(\mathbf{I})$. Since $\mathbf{I}$ was arbitrary, this shows that Q and Q' are equivalent. $\square$

Now, it is straightforward to show by induction that all repeated occurrences of a head variable can be expressed by equalities between distinct variables. It remains to show that no other inclusions hold.
To prove that claim it remains to show that all inclusions are strict and that for any pair of classes C, C' that are not related by an inclusion there exist queries $Q \in C \setminus C'$ and $Q' \in C' \setminus C$. The following trivial lemma will be useful for our proof, since it allows us to exploit inclusions to conclude non-inclusions from other non-inclusions.

Lemma 2. *Let A, C, C', B be sets such that $A \subseteq C$, $C' \subseteq B$. Then*

$$C' \nsubseteq C \qquad \textit{implies} \qquad C' \nsubseteq A \quad \textit{and} \quad B \nsubseteq C.$$

As a consequence, we only have to prove some crucial non-inclusions, from which others will follow. Of course, we get the best leverage of the lemma if we prove non-inclusions "$C' \nsubseteq C$" for sets C', C, where C' has many supersets and C has many subsets.
One readily checks that it is sufficient to show the following non-inclusions:

- $\mathsf{CQ}_{\mathit{rep}} \not\subseteq \mathsf{CQ}_{\mathit{const}}$
- $\mathsf{CQ}_{\mathit{const}} \not\subseteq \mathsf{CQ}_{=}$
- $\mathsf{CQ}_{=} \not\subseteq \mathsf{CQ}_{\mathit{rep,const}}$.

Our non-inclusion proofs will all follow the same pattern. We identify a property P for which we show that all queries in C satisfy P. Then we identify a query in C' that does not have this property.

Proposition 3. *For every binary query $Q \in \mathsf{CQ}_{\mathit{const}}$ one of the two following statements holds:*

- $|Q(\mathbf{I})| \leq 1$ *for all instances* $\mathbf{I}$*;*
- *there exist constants a, b with $a \neq b$ and an instance $\mathbf{I}$ such that $(a, b) \in Q(\mathbf{I})$.*

Corollary 4. $\mathsf{CQ}_{\mathit{rep}} \not\subseteq \mathsf{CQ}_{\mathit{const}}$.

Proof. The query $Q(x, x) \text{ :– } r(x)$ is in $\mathsf{CQ}_{\mathit{rep}}$, but does not have the properties mentioned in Proposition 3. □

Proposition 5. *For every query $Q \in \mathsf{CQ}_{=}$ and every instance $\mathbf{I}$ we have that a constant c occurs in $Q(\mathbf{I})$ only if c occurs in $\mathbf{I}$.*

Corollary 6. $\mathsf{CQ}_{\mathit{const}} \not\subseteq \mathsf{CQ}_{=}$.

Proof. The query $Q(a) \text{ :– } r(b)$ is in $\mathsf{CQ}_{\mathit{const}}$, but does not have the above property. □

Proposition 7. *All queries in $\mathsf{CQ}_{\mathit{rep,const}}$ are satisfiable.*

Corollary 8. $\mathsf{CQ}_{=} \not\subseteq \mathsf{CQ}_{\mathit{rep,const}}$.

Proof. The query $Q() \text{ :– } r(a),\ a{=}b$ is in $\mathsf{CQ}_{=}$ and is not satisfiable. Because of Proposition 7, the query Q is not in $\mathsf{CQ}_{\mathit{rep,const}}$. □

This proves the claim.