1. (3 points) Prove that it is undecidable whether a CALC query is monotonic (recall that a query \( Q \) is monotonic if for all databases \( D_1, D_2 \), if \( D_1 \subseteq D_2 \) then \( Q(D_1) \subseteq Q(D_2) \)).

2. (3 points) Let \( Q_1, Q_2, Q_3 \) be conjunctive queries (no equality). Prove that, if \( Q_1 \subseteq (Q_2 \cup Q_3) \), then \( Q_1 \subseteq Q_2 \) or \( Q_1 \subseteq Q_3 \) (recall that the notation \( P \subseteq R \) for queries \( P, R \) means that \( P(I) \subseteq R(I) \) for every database \( I \)).

3. (i) (4 points) The language \( \exists^* \forall^* \text{CALC} \) consists of all CALC sentences of the form

\[
\exists y_1 \ldots \exists y_n \forall z_1 \ldots \forall z_m \varphi(y_1, \ldots, y_n, z_1 \ldots z_m)
\]

where \( \varphi \) is a quantifier-free CALC formula. Prove that satisfiability is decidable for sentences in \( \exists^* \forall^* \text{CALC} \). \textbf{Hint}: Come up with a bound on the size of databases one needs to look at in order to check satisfiability. More precisely, show that a sentence \( Q \) in this language is satisfiable iff it is satisfied on some database of size bounded by \( f(Q) \) for some computable function \( f \).

(ii) (3 points) The language \( CQ^- \) consists of queries of the form \( \varphi(\bar{x}) - \psi(\bar{x}) \) where \( \varphi \) and \( \psi \) are CQs. Prove that equivalence of \( CQ^- \) queries is decidable.

4. (5 points) A database consists of the following binary relations:

<table>
<thead>
<tr>
<th>P</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>R</td>
<td>A</td>
<td>C</td>
</tr>
</tbody>
</table>

(\( P \) with attributes \( AB \), \( Q \) with attributes \( BC \) and \( R \) with attributes \( AC \)). Consider the relational algebra query \( P \bowtie Q \bowtie R \) (equivalent to the formula \( \varphi(a, b, c) = P(a, b) \land Q(b, c) \land R(a, c) \)). Assuming that \( P, Q \) and \( R \) have size at most \( n \) (i.e. they each contain at most \( n \) pairs), find the tightest upper bound you can on the size of \( P \bowtie Q \bowtie R \), as a function of \( n \).

\textbf{Hint}: There is an obvious upper bound of \( n^2 \). Try to do better.
5. (2 points) Recall the movie database in Problem 1 of the previous homework, and the query *List the theaters showing only movies by Hitchcock.* Express this query in nr-Datalog\(^{\sim}\).