Take home final exam  
CSE 233, Spring 2022

Please email the pdf to me (vianu@cs.ucsd.edu) as soon as you are done, but no later than **Sunday, June 5, 11:59pm**. This is an **individual** exam, please do not collaborate. Results proven in class, text, or homeworks, may be used without proof. External sources may **not** be used (if in doubt, please ask). **Most problems have very simple solutions if one makes use of known results.**

*Elegance and conciseness will be taken into account.*

**Problem 1** (5 points) Let $R$ be a relation with attributes ABCD. Reduce the number of joins in the algebra query

$$\pi_{AB}[\pi_{BCD}(R) \bowtie \pi_{ACD}(R)] \bowtie \pi_{AD}(R)$$

knowing that the query is only applied to databases satisfying the dependencies:

$$B \rightarrow D, D \rightarrow C, A \rightarrow B.$$ 

Start by writing a CQ equivalent to the above algebra query, then chase and minimize its body, and finally convert the result back to an algebra query.

**Problem 2** (4 points) Show that there is no algorithm that, given a relational algebra query, returns an equivalent query with a minimum number of joins.

**Problem 3** (10 points) A database holds data about courses in an undergraduate CSE program, prerequisites, and records for each student. The tables are the following:

<table>
<thead>
<tr>
<th>Course</th>
<th>C-id</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record</td>
<td>S-id</td>
</tr>
<tr>
<td>Prereq</td>
<td>C-id</td>
</tr>
</tbody>
</table>

In relation Prereq, P-id provides a prerequisite for C-id (assume this is acyclic). Relation Record provides, for each student S-id, the courses C-id he/she has already taken. Assume that Record[C-id] $\subseteq$ Course[C-id], Prereq[C-id] $\subseteq$ Course[C-id] and Prereq[P-id] $\subseteq$ Course[C-id]. Consider the following query:

“For each student, find the set of courses whose prerequisites they have already taken”

Express this query in the following languages: (i) CALC (ii) relational algebra (iii) nr-Datalog$^{-}$ (iv) inflationary Datalog$^{-}$, and (v) semi-positive Datalog$^{-}$, assuming also given a successor relation $\text{succ}$ on course ids, as well as constants $\text{min}$ and $\text{max}$ providing the minimum, resp. maximum course id.

**Problem 4** (5 points) Consider a binary relation $\text{Child}$ where $\text{Child}(a,b)$ means that $b$ is the child of $a$ (assume $\text{Child}$ is acyclic). (i) Write a Datalog program
computing the set of pairs \((c, d)\) where \(c\) and \(d\) have a common ancestor and are of the same generation with respect to this ancestor. (ii) Is it possible to write such a Datalog program if \(c\) and \(d\) are additionally required to be distinct?

**Problem 5** (6 points) Let \(G\) be a binary relation representing the edges of a directed graph. Consider the set \(G_1\) of directed graphs \(G\) consisting of a single cycle and the set \(G_{\geq 1}\) of directed graphs \(G\) consisting of a union of any number of disjoint cycles. (i) (3 points) Does \(\text{while}^+\) express all \(\text{PTIME}\) queries on \(G_1\)? (ii) (3 points) Does \(\text{while}^+\) express all \(\text{PTIME}\) queries on \(G_{\geq 1}\)?

**Problem 6** (3 points) We know that Datalog queries are monotonic and can be evaluated in \(\text{PTIME}\) (with respect to the database). Does Datalog express all the monotonic \(\text{PTIME}\) queries?

**Problem 7** (3 points) Prove that it is undecidable, given a \(\text{while}^+\) program, whether there exists a \(\text{CALC}\) query equivalent to it.