Exercise Sheet 3: Complexity of FO Query Answering

Jonas Karge, Sebastian Rudolph

Database Theory, 2021-04-30, Summer Term 2021

Exercise 3.1. Three decision problems related to query answering have been introduced in the lecture:

- Boolean query entailment,
- query answering, and
- the query emptiness problem.

Show that these problems are equivalent, i.e., if we have an algorithm that solves any of these problems, then we can also use it to solve the others.

Hint:

Assume we have a Turing Machine $\mathcal{M}_{BQE}$ solving Boolean Query Entailment. How can we use this machine to solve Query Answering?

Exercise 3.2.

Elaborate on why FO query evaluation is still PSpace-complete if we consider a QBF in negation normal form to reduce from.

Exercise 3.3. Expressions of relational algebra under named perspective can be translated into Boolean circuits, in a similar fashion to the translation illustrated for FO queries in the lecture. Show how each operator of relational algebra gives rise to a corresponding circuit by describing the circuits for the following expressions:

1. $\sigma_{n=c}(R)$ where c is a constant
2. $\sigma_{n=m}(R)$ where m is an attribute
3. $\pi_{a_1,\ldots,a_n}(R)$
4. $R \bowtie S$
5. $\delta_{a_1,\ldots,a_n \to b_1,\ldots,b_n} R$
6. $R - S$
7. $R \cup S$
8. $R \cap S$

Exercise 3.4. It was claimed in the lecture that the composition of two functions that can each be computed in LogSpace can also be computed in LogSpace. How can this be achieved, considering the fact that the output of one LogSpace function may already require more than logarithmic space?

Hint:

We will need two logspace-computable functions (f and g) and two logspace transducers $\mathcal{M}_f$ and $\mathcal{M}_g$ computing those functions. We want to show that $f \circ g$ is also logspace-computable. What would be a straightforward approach and why doesn't it work?

Exercise 3.5. Is the question "P = NP?" decidable? Explain your answer by either showing that there is an algorithm that correctly answers this question, or by showing that such an algorithm cannot exist.