Part 1: Consistent Query Answering

Question 1.1: Conjunctive Query Answering

Consider the Boolean CQ \( Q() : R(z, y), S(y, x, z), T(z, w), U(x) \).

1. Using the trichotomy theorem of Koutris and Wijsen (studied in class), show that \( \text{Consistent}^Q \Sigma \) cannot be phrased in Relational Calculus.

2. Devise a polynomial-time algorithm for computing \( \text{Consistent}^Q \Sigma (I) \) on a given input instance \( I \). (Hint: adapt the strategy for \( R(z, y), S(y, x) \) we saw in class.)

Question 1.2: Aggregate Query Answering

Consider a schema \( (S, \Sigma) \) where \( S \) consists of a single relation schema \( R(A, B) \). Consider the following aggregate query.

\[ Q: \text{SELECT} \max(B) \text{ FROM} R \]

Given an inconsistent instance \( I \) over \( (S, \Sigma) \), we are interested in the extremal (minimal and maximal) values:

\[ Q_{\min}(I) \overset{\text{def}}{=} \min \{Q(J) \mid J \in \text{Repairs}_\Sigma(I)\} \]
\[ Q_{\max}(I) \overset{\text{def}}{=} \max \{Q(J) \mid J \in \text{Repairs}_\Sigma(I)\} \]

Devise polynomial-time algorithms for computing \( Q_{\min}(I) \) and \( Q_{\max}(I) \) for the following different sets \( \Sigma \) of constraints:

1. \( \Sigma = \{A \rightarrow B\} \)
2. \( \Sigma = \{A \rightarrow B, B \rightarrow A\} \)

Note that, in total, you are requested to devise four algorithms. (Hint: three of the algorithms are straightforward, and one is more intricate.)

Part 2: Knowledge Compilation

This part is a preparation for the lectures on probabilistic query answering and the next (final) assignment.
Question 2.1: Terminology

From the article “A Knowledge Compilation Map” by Darwiche and Marquis [1], find and provide a definition, in your own words, for each of the following terms:

1. Negation Normal Form (NNF);
2. Decomposable Negation Normal Form (DNNF);
3. Deterministic Decomposable Negation Normal Form (d-DNNF);

Question 2.2: Satisfiability Testing

Show a polynomial-time algorithm for deciding on the satisfiability (i.e., existence of a satisfying assignment to the variables) for given DNNF and d-DNNF circuits.

Question 2.2: Model Counting

1. Show a polynomial-time algorithm for model counting (i.e., computing the number of satisfying assignments to the variables) of a given d-DNNF circuit.
2. Prove that no polynomial-time model counting exists for the class of DNNF circuits, or else \( P = NP \).

Good luck!

References