5. Datalog

Instructions: Work in groups of 2 students. You can write up your answers by hand (provided your handwriting is legible) or use a word processing system like Latex or Word. Note that experience has shown that Word is in general difficult to use for this kind of task. If you prefer to write up your solution by hand, submit a scanned electronic version. Please, include name and email address in your submission.

1. Properties of Datalog Programs

Suppose P is a property of graphs, definable by a datalog program. Show that P is preserved under extensions and homomorphisms. That is, if G is a graph satisfying P, then

1. every supergraph of G satisfies P
2. if h is a graph homomorphism, then $h(G)$ satisfies P.

(8 Points)

2. Definability in Datalog

Which of the following graph properties is definable by a datalog program?

1. There is a trivial cycle (a trivial cycle consists of a single node a and an edge $\langle a, a\rangle$).
2. There is a nontrivial cycle.
3. There exist two nodes a and b such that there is a path between a and b.
4. There is a path between any two nodes a and b.
5. The number of nodes is even.
6. There is a Hamiltonian path.

(6 Points)

3. Queries in datalog$^\neg$

Consider a database for metro and bus stations, with two relations

$$\texttt{metro}(\texttt{station}, \texttt{next\_station})$$
$$\texttt{bus}(\texttt{station}, \texttt{next\_station}).$$

Write stratifiable datalog$^\neg$ programs that answer the following queries:

1. Find the terminal bus stations.
2. Find the pairs (a, b) of stations such that one can go from a to b by metro, but not by bus.
3. Find the pairs (a, b) of stations such b can be reached from a by some combination of metro and/or bus, but not by metro or bus alone.

A pure metro route is a path $a = s_0, s_1, \ldots, s_n = b$, $n > 0$, such that each step (s_i, s_{i+1}) is in `metro` and there is no path from s_i to s_{i+1} by bus, for all $0 \leq i < j \leq n$.

4. Find the pairs (a, b) of stations such that there is a pure metro route from a to b.

Consider an extension of the previous database, where both relations have an additional attribute `duration`, such that tuples (a, b, d) represent that going from a to the next station b takes d minutes.

5. Write a datalog$^\neg$ program, using the built-in predicate $<$ on durations, which computes a 3-ary relation

   $$\texttt{longestLink}(\texttt{from}, \texttt{to}, \texttt{duration})$$

   all tuples (a, b, d) such that b is directly reachable from a in time d, but not shorter, and d is maximal for all such b, that is, d is the the longest time to directly reach any b' from a.

Test all programs in DLV on a sample database. Submit the tested DLV code.

(10 Points)

4. Transitive Closure and First-order Logic

Is it possible to express transitive closure in first-order logic (with the classical logical semantics that allows for infinte interpretations)? In other words, given a binary relation $R(\cdot, \cdot)$, is it possible to write a formula $\varphi(x, y)$ such that in every interpretation of R, the formula φ is satisfied exactly by those pairs of domain elements that are in the transitive closure R^+ of R?
Hint 1: Exploit the fact that the compactness theorem holds for first-order logic.

Hint 2: Consider the following statements about a point a and their formulation as logical formulas:

- There exists a point reachable from a that is an R-sink, that is, no R-edge is emanating from that point.
- For each natural number k: All points reachable from a via a path of length k have an emanating R-edge.

(6 Points)