Deductive Databases
Summer Term 2018
Prof. Dr. W. May

2. Unit: Datalog

Discussion by 6./8.6.2018

Exercise 1 (Äquivalenz von Algebra und Datalog) Show that for every expression of the relational algebra there is an equivalent stratified Datalog program.

Exercise 2 (Datalog to Algebra)
Consider the translation of Datalog programs with a distinguished answer predicate to the relational algebra.

- Given a rule $B \leftarrow C_1 \wedge \ldots \wedge C_m \wedge \neg D_{m+1} \wedge \ldots \wedge \neg D_{m+n}$
  where the C_i and D_i are of the form $R_i(a_1, \ldots, a_\ell)$, a_j constants or variables. Give an algebra expression that returns the relation defined by it.
- Which additional constructs must also be translated?
- Consider the following program:

```
res(X,Z) :- v(X,_,_Y), q(_,_Y,Z), ¬r(Z,_).
res(X,Z) :- v(X,_Y,Z), ¬r(_Y,_), ¬w(X).
v(X,Y,Z) :- p(Z,_,X), q(X,Y,_).
v(X,Y,Z) :- p(X,Y,Z), Y<4.
w(X) :- s(_,X), t(X,_).
```

  where p/3, q/3, r/2, s/2, t/2 are EDB relations, `v/3, w/1` are IDB relations (views).
  Give the algebra expression that corresponds to the res predicate.

Exercise 3 (Stratified Datalog) Consider a stratified Datalog program P with strata P_1, P_2, $\ldots$, P_n and the following definitions:

a)

$$\begin{aligned} \mathcal{J}_0 &= \emptyset \\ \mathcal{J}_k &= \mathcal{J}_{k-1} \cup T^{\omega}_{P_k}(\mathcal{J}_{k-1}) \quad \text{for } 1 \leq k \leq n \\ \mathcal{S}(P) &= \mathcal{J}_n \end{aligned}$$

b)

$$\begin{aligned} \mathcal{J}_0 &= \emptyset \\ \mathcal{J}_k &= T^{\omega}_{P_1 \cup \ldots \cup P_k}(\mathcal{J}_{k-1}) \quad \text{for } 1 \leq k \leq n \\ \mathcal{S}'(P) &= \mathcal{J}_n \end{aligned}$$

Which of them are equivalent to the stratified semantics? (give a proof sketch or a small counterexample).

Exercise 4 (Stratified Datalog)

Give an example for the nonmonotonicity of the stratified semantics,

show that for a stratifiable program P there can be multiple minimal models.

Exercise 5 (Datalog-Anfragen an Mondial: Schweizer Sprachen) Give Datalog programs for the following queries against the Mondial database. Compare with the same queries in the algebra and in the relational calculus.

a) All codes of countries in which some language is spoken that is also spoken in Switzerland.
b) All codes of countries in which only languages are spoken that are not spoken in Switzerland.
c) All codes of countries in which only languages are spoken that are also spoken in Switzerland.
d) All codes of countries in which all languages are spoken that are spoken in Switzerland.

Exercise 6 (Datalog-Anfragen an Mondial: Landlocked)

- Give a Datalog program that returns the names of all countries that have no coast.
- Give a Datalog program that returns the names of all countries that have no coast and that have no neighbor country that has any coast.
- Give the dependency graph of your program.

Exercise 7 (Aggregation in Datalog/XSB) Define the aggregation operators in XSB in a module `aggs.P`.

The syntax of the comparison predicates and of the arithmetic operators is given in Sections 3.10.5 (Inline Predicates) and 4.3 (Operators) of the XSB Manual Part I.

Then use `aggs.P` for answering the following queries in Datalog:

a) Give for each country the name and the number of neighbors.
b) Give the name of the country that has the highest number of neighbors (and how many).
c) Give the average area of all continents (to test avg).
d) Give the average latitude and longitude of all cities.