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

1. Unit: Kalkül I

Self-contained subformulas (i.e., formulas in RANF) will also be needed for translating complex queries into Datalog programs. The transformation into RANF works in the same intuitive way as done in the DB lecture before knowing the relational calculus at all. For complex cases (and for the computer), a systematic formal procedure helps.

Exercise 1 (Kalkül: Sichere, Wertebereichsunabhängige Anfragen) Check for the following queries whether they are in SRNF (give $rr(G)$ for each of their subformulas).

For the formulas that are in RANF:

- check whether the formulas are in RANF. If not, give an equivalent formula in RANF.
- Give equivalent expressions in the relational algebra and in SQL (develop the SQL expressions both from the original formula and from the RANF formula).

a) $F(X, Y, Z) = p(X, Y) \wedge (q(Y) \vee r(Z))$,
b) $F(X, Y) = p(X, Y) \wedge (q(Y) \vee r(X))$,
c) $F(X, Y) = p(X, Y) \wedge \neg\exists Z : r(Y, Z)$,
d) $F(X) = p(X) \wedge \exists Y : (q(Y) \wedge \neg r(X, Y))$,
e) $F(X) = p(X) \wedge \neg\exists Y : (q(Y) \wedge \neg r(X, Y))$
f) $F(X, Y) = \exists V : (r(V, X) \wedge \neg s(X, Y, V)) \wedge \exists W : (r(W, Y) \wedge \neg s(Y, X, W))$

Exercise 2 (Relationale Anfragen an Mondial: Schweizer Sprachen) Give expressions in the relational calculus for the following queries against the Mondial database. Compare with the same queries in the relational Algebra and in SQL.

a) All codes of countries, in which some languages 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 that are spoken in Switzerland are also spoken.

Exercise 3 (RANF to Algebra – Minus) Give expressions in the relational algebra and in the relational calculus for the query *"Full names of all countries that have more than 1000000 inhabitants and are not member of the EU"*.

Check whether the calculus expression is in SRNF and RANF, and transform it into the relational algebra. Compare the result with the algebra expression.

Exercise 4 (Division: Äquivalenz von Algebra und Kalkül) For the relational algebra, the division operator has been introduced as a derived operator (cf. lecture "Databases"). Consider the relation schemata $r(A, B)$ and $s(B)$.

$$r \div s = \{\mu \in Tup(A) \mid \{\mu\} \times s \subseteq r\} = \pi[A](r) \setminus \pi[A]((\pi[A](r) \times s) \setminus r).$$

Derive a query in the relational calculus from the left-hand side, and prove the equivalence with the right-hand side.

Exercise 5 (Kalkül: Gruppierung und Aggregation) Define a syntactical extension for the relational calulus, that allows to use aggregate functions similar to the `GROUP BY` functionality of SQL.

For this, consider only aggregate functions as simple applications over single attributes like max(population), but not more complex expressions like max(population/area).

- What is the result of an aggregate function, and how can it be used in the calculus?
- Which inputs does an aggregate function have?
- how can this input be obtained from the database?

Give a calculus expression for the query "For each country give the name and the total number of people living in its cities".

Exercise 6 (Kalkül→Algebra) Consider the relation schemata $R(A, B)$, $S(B, C)$ und $T(A, B, C)$.

a) Give an equivalent algebra expression for the following safe relational calculus expression:

$$F_1(X, Y) = T(Y, a, Y) \wedge (R(a, X) \vee S(X, c)) \wedge \neg T(a, X, Y)$$

   Proceed as shown in the lecture for the equivalence proof.

b) Simplify the expression.

c) Extend the expression from (a) to

$$F_2(Y) = \exists X : (F_1(X, Y) \wedge X > 3)$$