**Datenbanktheorie**
**Sommersemester 2012**
Prof. Dr. W. May

# 2. Übungsblatt: Kalkül II

Besprechung voraussichtlich am 24./25.5.2012

**Aufgabe 4 (Division: Äquivalenz von Algebra und Kalkül)** Für die relationale Algebra wurde der Divisions-Operator eingeführt (vgl. Vorlesungsfolien). Gegeben sind die Relationsschemata $r(A, B)$ und $s(B)$.

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

Leiten Sie aus dem linksseitigen Ausdruck eine Anfrage des relationalen Kalküls ab, und beweisen Sie die Äquivalenz mit dem rechtsseitigen Ausdruck.

---

Der linksseitige Ausdruck: Die Menge aller möglichen Tupel über $A$ beschreibt man mit $F(X) = ADOM(X)$. Der Rest ist dann einfach: für alle Werte $Y$ in $S$ muss die Kombination aus dem jeweiligen $X$ und $Y$ in $R$ sein:

$$F(X) = ADOM(X) \wedge \forall Y : (s(Y) \rightarrow r(X, Y)) .$$

Jetzt kann man sich gleich überlegen, dass man anstatt $ADOM(X)$ auch gleich auf die möglichen $A$-Werte von $R$ einschränken kann:

$$F(X) = \exists Z : r(X, Z) \wedge \forall Y : (s(Y) \rightarrow r(X, Y)) .$$

Die Anfrage ist nicht in SRNF. Sie ist äquivalent zu

$$F(X) = \exists Z : r(X, Z) \wedge \neg\exists Y : (s(Y) \wedge \neg r(X, Y)) .$$

Diese ist in SRNF (also domain-unabhängig, womit man sie für theoretische Untersuchungen verwenden kann), aber nicht in RANF.

Umformung in RANF ("push-into-not-exists):

$$F(X) = \exists Z : r(X, Z) \wedge \neg\exists Y : (\exists Z_2 : r(X, Z_2)) \wedge s(Y) \wedge \neg r(X, Y))$$

Ableitung des entsprechenden Algebra-Ausdruckes:

$p(X, Y) \wedge (q(Y) \vee r(Z))$:

| $F$ | *Algebra* |
|---|---|
| $(\exists Z : r(X, Z)) \wedge s(Y) \wedge \neg r(X, Y)$ | $(\pi[A](r) \times s) \setminus r$ |
| $\exists Y : (\exists Z : r(X, Z)) \wedge s(Y) \wedge \neg r(X, Y)$ | $\pi[A]((\pi[A](r) \times s) \setminus r)$ |
| | (der Ausdruck hat das Format $A$) |
| $\exists Z : r(X, Z)$ | $\pi[A](r)$ (hat auch das Format $A$) |
| $F(X)$ wie oben | $\pi[A](r) \setminus \pi[A]((\pi[A](r) \times s) \setminus r)$ |

... ist exakt das was oben gesucht war.

---

**Aufgabe 5 (Kalkül: Gruppierung und Aggregation)** Definieren Sie eine Syntaxerweiterung für den relationalen Kalkül, die es erlaubt, Aggregatfunktionen mit der aus SQL bekannten Funktionalität von `GROUP BY` zu verwenden.

Betrachten Sie dabei für die Aggregatfunktionen nur einfache Anwendungen auf Attribute, wie z.B. max(population), nicht aber komplexere Ausdrücke wie max(population/area).

- Was ist das Ergebnis einer Aggregatfunktion, und wie verwenden Sie es im Kalkül?
- Welche direkte Eingabe hat die Aggregatfunktion?
- Wie wird diese Eingabe aus der Datenbank gewonnen?

Geben einen Kalkül-Ausdruck für die Anfrage "Geben Sie für jedes Land den Namen sowie die Anzahl der Stadtbevölkerung an" an.

---

Das Ergebnis ist eine Zahl. Man kann sie an eine Variable binden, oder in einem Vergleich verwenden. Die Aggregatfunktion ist daher wie ein Term (dessen Auswertung einen Wert ergibt) zu betrachten.

Die direkte Eingabe der Aggregatfunktion ist eine Menge/Liste von Werten, über die aggregiert (summiert, gezählt, ...) wird.

Diese Menge/Liste kann man als Ergebnisse einer Formel (ggf. korrelierte Subquery) mit einer freien Variablen erhalten.

Die Ergebnisse werden nach einer oder mehreren der in der Subquery ebenfalls freien Variablen gruppiert. Diese treten meistens in weiteren Literalen ausserhalb der Aggregation auch auf.

$$X = \textsf{\textit{agg-op}}\{\textsf{\textit{var}}\ [\textsf{\textit{group-by-vars}}]; \textsf{\textit{subq-fml}}\}$$

wobei in *subq-fml* die *group-by-vars* sowie *var* frei auftreten. Z.B.

$$\begin{array}{l} F(CN, SumCityPop) = \\ \qquad \exists C, A, P, Cap, CapProv : country(CN, C, A, P, Cap, CapProv) \land \\ \qquad SumCityPop = \mathsf{sum}\{CityPop\ [C]; \\ \qquad\qquad\qquad \exists CtyN, CtyProv, L1, L2 : city(CtyN, CtyProv, C, CityPop, L1, L2\} \end{array}$$

gruppiert nach $C$, bildet die Summe über $CityPop$ und bindet den Wert an $SumCityPop$.

Hinweis:

- eine ähnliche Syntax wird in F-Logic verwendet;
- das Vorgehen in XSB ist genauso, der Benutzer muss aber mehr ausprogrammieren:
  - die Liste kann mit dem üblichen Prolog-Prädikat "findall" oder "bagof" erzeugt werden;
  - die Aggregationsoperation über die Liste muss mit der Prolog-üblichen Behandlung einer Liste ausprogrammiert werden.

---

**Aufgabe 6 (Algebra→Kalkül)** Gegeben sind die Relationsschemata $R(A, B)$, $S(B, C)$ und $T(A, B, C)$.

Geben Sie zu dem folgenden Algebraausdruck einen äquivalenten sicheren Kalkülausdruck an:

$$(\pi[A, B]((R \bowtie S) - T)) \cup R$$

---

Am besten fängt man von unten an (um die Variablen zu kennen, die jeweils gebunden sind):

| *Algebra* | *Calc* | $free(\_)$ | $rr(\_)$ |
|---|---|---|---|
| $R \bowtie S$ | $R(A,B) \wedge S(B,C)$ | $A,B,C$ | $A,B,C$ |
| $(R \bowtie S) - T$ | $(R(A,B) \wedge S(B,C)) \wedge \neg S(A,B,C)$ | $A,B,C$ | $A,B,C$ |
| $\pi[A,B]((R \bowtie S) - T))$ | $\exists C : (R(A,B) \wedge S(B,C)) \wedge \neg T(A,B,C)$ | $A,B$ | $A,B$ |
| $(\pi[A,B]((R \bowtie S) - T))) \cup R$ | $(\exists C : (R(A,B) \wedge S(B,C)) \wedge \neg T(A,B,C)) \vee R(A,B)$ | $A,B$ | $A,B$ |

Der Ausdruck ist in SRNF und in RANF (und sicher).

**Aufgabe 7 (Kalkül→Algebra)** Gegeben sind die Relationsschemata $R(A,B)$, $S(B,C)$ und $T(A,B,C)$. Geben Sie zu dem folgenden sicheren Kalkülausdruck einen äquivalenten Algebraausdruck an:

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

Gehen Sie dabei zuerst so vor, wie in der Vorlesung im Beweis der Äquivalenz angegeben. Vereinfachen Sie dann den Ausdruck.

Zuerst jede der drei Teilformeln in der Konjunktion (als $F_2, F_1$ und $F_3$ jeweils genannt) getrennt betrachten:

Dem Literal $F_1(Y) = T(Y,a,Y)$ entspricht der Teilausdruck

$$E_1 = \rho[A \to Y](\pi[A](\sigma[(A = C) \wedge (B = a)](T))) \ .$$

Der Teilformel $F_2(X) = R(a,X) \vee S(X,c)$ entspricht der Teilausdruck

$$E_2 = \rho[B \to X](\pi[B](\sigma[A = a](R))) \cup \rho[B \to X](\pi[B](\sigma[C = c](S))) \ .$$

Negiertes Literal $F_3(X,Y) = \neg T(a,X,Y)$: Dem Literal $F_4(X,Y) = T(a,X,Y)$ entspricht der Teilausdruck

$$E_4 = \rho[B \to X, C \to Y](\pi[B,C](\sigma[A = a](T)))$$

Nach dem Vorgehen in der Vorlesung ist der $F_3(X,Y)$ entsprechende Ausdruck dann

$$E_3 = E^2 - \rho[B \to X, C \to Y](\pi[B,C](\sigma[A = a](T)))$$

wobei $E^2 = ((\pi[A](R) \cup \pi[B](R) \cup \pi[B](S) \cup \pi[C](S) \cup \pi[A](T) \cup \pi[B](T) \cup \pi[C](T)) \times (\pi[A](R) \cup \pi[B](R) \cup \pi[B](S) \cup \pi[C](S) \cup \pi[A](T) \cup \pi[B](T) \cup \pi[C](T)))$ alle 2-Tupel aus Werten aus der DB enthält.

Damit ist $E = E_1 \bowtie E_2 \bowtie (E^2 - E_4)$ der äquivalente Algebraausdruck.

Vereinfachen: $E_1$ und $E_2$ haben keine Variable/Spalte gemeinsam, also kann man zu $(E_1 \times E_2) \bowtie (E^2 - E_4)$ vereinfachen. Beide Teilterme binden $X$ und $Y$, so dass man das $E^2$ auch einfach weglassen kann und $(E_1 \times E_2) - E_4$ erhält.