BLATT 9

DANIEL SCHMIDT & PAMELA FLEISCHMANN

Aufgabe 1. Um zu zeigen, dass Datalog ohne Rekursion, aber mit Negation und sicheren Regeln die gleiche Ausdruckskraft wie die Relationenalgebra hat muss gezeigt werden, dass $\mu(L) = \mu'(L')$ gilt, wobei Datalog als (L, μ) definiert ist und die Relationenalgebra (L', μ') ist. Hierzu gilt zu zeigen, dass dies für ein beliebiges Datenbankschema σ gilt. Also müssen folgende Aussagen gezeigt werden:

$$1)\forall e' \in L' : \exists e \in L : \mu'(e') = \mu(e)$$
$$2)\forall e \in L : \exists e' \in L' : \mu(e) = \mu'(e')$$

Diese werden nun einzelnd gezeigt:
1). Um zu zeigen, dass jeder Ausdruck der in der Relationenalgebra in einen bedeutungsgleichen Ausdruck in Datalog umgeformt werden kann muss lediglich gezeigt werden, dass die einzelnen Konzepte jeweils ausgedrückt werden können. Die Zusammensetzung dieser ist implizit durch die Zusammensetzbarkeit der Ausdrücke gegeben. Seien e_{Rel} und e'_{Rel} im Folgenden Ausdrücke der Relationenalgebra; Seien e_{Dat} und e'_{Dat} bereits überführte Datalog Regeln und e''_{Dat} die durch die Umformung entstehende Regel.

Selektion. Sei $e_{Rel} = e'_{Rel}[es_1, \cdots, es_n]$ mit $n > 0$ und es_i Vergleichsausdrücke mit $0 < i \leq n$. Dann lässt sich e in Datalog als $e''_{Dat}(V) : -e'_{Dat}(V), es_1(V), \cdots, es_n(V)$ ausdrücken, wobei V die Liste der Argumente, bzw. die Spalten der Tabelle sind; Beide Aussagen sind äquivalent.

Umbenennung. Sei V die Liste aller Spalten in e_{Rel} und V' die Liste aller Spalten in $e_{Rel}[k \to j]$, so gilt $k \in V \land k \notin V' \land j \notin V \land j \in V'$. Dann lässt sich die Umbenennung darstellen als $e''_{Dat}(V) : -e_{Dat}(V')$..

Projektion. Eine Projektion lässt sich analog zur Umbenennung umsetzen, allerdings muss hierbei die Menge V' so gewählt werden, dass jedes $k \in V$ durch ein "_" ersetzt wird das nicht in der Menge der erlaubten Felder enthalten ist.

Kartesisches Produkt. Sei der Ausdruck $e''_{Rel} = e_{Rel} \times e'_{Rel}$ gegeben. Seien die Parameter $Param_{e_{Rel}}, Param_{e'_{Rel}}, Param_{e''_{Rel}}$ wie folgt definiert:

Sei $Param_{e_{Rel}}$ gegeben als alle Spalten im Resultat von e und $Param_{e'_{Rel}}$ als allen Spalten im Resultat von e'. Dann ist $Param_{e''_{Rel}}$ zu definieren als Liste von Parametern $x_1, \cdots x_n, y_1, \cdots, y_m$ mit $n = |Param_{e_{Rel}}|, \quad m = |Param_{e'_{Rel}}|, \quad x_i \in Param_{e_{Rel}}$ für $1 \leq i \leq n, \quad y_i \in Param_{e'_{Rel}}$ für $1 \leq i \leq m$.
In Prolog lässt sich das kartesische Produkt dann durch
$e''_{Dat}(Param_{e''}) : -e_{Dat}(Param_e)), e'_{Dat}(Param_{e'}))$ ausdrücken.

Differenz. Sei der Ausdruck $e''_{Rel} = e_{Rel} \backslash e'_{Rel}$ gegeben. Sei der Parameter $Param_{e_{Rel}}$ definiert als alle Spalten im Resultat von e_{Rel}, [1] so lässt sich die Differenz definieren als Regel

$e''_{Dat}(Param_e)$:− $e(Param_e)$, not $e'(Param_e)$.

Vereinigung. Sei der Ausdruck $e''_{Rel} = e_{Rel} \cup e'_{Rel}$ gegeben und V die Liste aller Spalten in e_{Rel} und damit auch in e'_{Rel}, so ist die äquivale Datalog Regel gegeben als:

e''_{Dat}(V) :− $e'_{Dat}(V)$.
e''_{Dat}(V) :− $e_{Dat}(V)$.

Weitere Operationen müssen nach Satz 2.1 nicht gezeigt werden, daher ist diese Richtung ausreichend bewiesen.
2). Analog zu 1) muss gezeigt werden, dass es zu jeder Operation in Datalog eine äquivalente in der Relationenalgebra gibt. Hierzu seien e_{Dat} und e'_{Dat} im Folgenden Ausdrücke in Datalog; Seien e_{Rel} und e'_{Rel} bereits überführte Ausdrücke der Relationenalgebra und e''_{Rel} der durch die Umformung entstehende Ausdruck.

Fakt. Fakten in Datalog lassen sich über das Konzept des Datenbankzustands in der Relationenalgebra abbilden, ist ein Fakt gegeben, so ist ein Eintrag in der entsprechenden Datenbanktabelle gegeben.

Und. Wie in Teil 1) zu sehen wird "und" bei diversen Operationen benötigt, somit sind auf der Rückrichtung ebenfalls diverse Fälle möglich. Hierbei lässt es sich am einfachsten anhand der Parametermenge V von e_{Dat} und V' von e'_{Dat} in der Regel $e''_{Dat} : -e_{Dat}, e'_{Dat}$ unterscheiden:

(1) $V = V'$: In diesem Fall handelt es sich um einen Schnitt von e_{Rel} udn e'_{Rel} in der Relationenalgebra.
(2) $V \cap V' = \emptyset$: Da beide Mengen disjunkt sind handelt es sich um ein kartesisches Produkt in der Relationenalgebra, was wie in 1) gezeigt eine äquivalente Darstellung ist.
(3) sonst: Da beide Mengen weder gleich noch disjunkt sind handelt es sich um ein Join, welches wie bereits gezeigt durch ein kartesisches Produkt und eine Selektion dargestellt werden kann.

[1]Äquivalent zu allen Spalten im Resultat von e'_{Rel} und allen Ergebnisspalten

Oder. Dies wurde bereits unter 1) - Vereinigung gezeigt.
Geänderte Variablenbezeichnungen im Regelkopf. Dies wurde bereits unter 1) - Umbenennung, Projektion gezeigt.
Vergleichsausdrücke. Dies wurde bereits unter 1) - Selektion gezeigt.
Not. Not lässt sich danke Closed World Assumption umschreiben als $(Tab_e \backslash e)[V]$, wobei Tab_e das kartesische Produkt aller in e vorkommenden extensionale Prädikate beschreibt und V die im Regelkopf von e verwendeten Variablen.

Aufgabe 2. ad. a)
Die äquivalente Darstellung in Datalog ist

```
a(A,B) :- R(A,B,_).
a(A,B) :- S(A,D), T(_,D,B).
```

ad. b)
Die äquivalente Darstellung in Datalog ist

```
sbt(A,B,C,G) :- S(A,C), T(B,C,G).
r'(A,B,C) :- R(A,B,C), A > B.
r'(A,B,C) :- R(A,B,C), C > 5.
rbs(A,B,C,G) :- r'(A,B,C), S(A,G).
q(A,B,C,G) :- sbt(A,B,C,G), not rbs(A,B,C,G).
```

Aufgabe 3. Zu einem Schema σ und einem zu σ passenden Ausdruck der Relationenalgebra existiert ein äquivalenter zu σ passender Ausdruck der Relationenalgebra, der als Operatoren nur die Vereinigung, die Differenz, die Selektion mit einfachen Vergleichsausdrücken, die Projektion, die Umbenennung und das kartesische Produkt enthält, sowie als Operationen nur Relationstypbezeichner aus σ und konstante DB-Relationen der Form $\{(A : c)\}$.

Beweis. Seien R und S Relationstypen über einer Attributmenge α. Definiere

$$
\begin{aligned}
R \cup S &:= \{t \in T_\alpha |\, t \in R \vee t \in S\},\\
R \cap S &:= \{t \in T_\alpha |\, t \in R \wedge t \in S\},\\
R \backslash S &:= \{t \in T_\alpha |\, t \in R \wedge t \notin S\},\\
R[\beta] &:= \{t|_\beta |\, t \in R\}.
\end{aligned}
$$

Sei vga ein Vergleichsausdruck und R ein Relationstyp über α. $t \models \text{vga}$ bezeichne, dass ein Tupel t den Vergleichsausdruck vga erfüllt (Auswerten des Tupels an den Stellen der Attributbezeichner). Definiere

$$R[\text{vga}] := \{t \in R |\, t \models \text{vga}\}.$$

Seien nun R und S Relationstypbezeichner über β resp. γ mit $\beta, \gamma \subseteq \alpha$ für eine Attributmenge α. Definiere

$$R \bowtie S := \{t \in T_{\beta\cup\gamma} |\, t|_\beta \in R \wedge t|_\gamma \in S\}.$$

Gilt $\beta \cap \gamma = \emptyset$, so schreibe $R \times S$ statt $R \bowtie S$. Betrachte nun $\alpha' \subseteq \alpha$ mit $\alpha'' = \{A_1, \dots, A_k\} \subseteq \alpha'$ und $\{B_1, \dots, B_k\} \cap \alpha' = \emptyset$ mit $\mathrm{dom}(B_i) = \mathrm{dom}(A_i)$ für alle $i \in [k]$ (hierdurch wird sichergestellt, dass in neue Attribute umbenannt wird). Definiere für $t \in T_{\alpha'}$

$$t[A_1 \to B_1, \dots, A_k \to B_k] =: s$$
$$\Leftrightarrow (\forall i \in [k] : s[B_i] = t[A_i]) \wedge t|_{\alpha' \setminus \alpha''} = s|_{\alpha' \setminus \alpha''}.$$

Darauf aufbauend sei

$$R[A_1 \to B_1, \dots, A_k \to B_k] := \{t[A_1 \to B_1, \dots, A_k \to B_k] \,|\, t \in R\}.$$

Betrachte nun wieder Relationstypen R und S über den Attributmengen β resp. γ mit $\beta, \gamma \neq \emptyset$. Für $S \neq \emptyset$ setze

$$R \div S := \{t \in T_{\beta \setminus \gamma} \,|\, \forall s \in S \exists r \in R : r|_{\beta \setminus \gamma} = t \wedge r|_\gamma = s\}.$$

Ein Tupel t ist genau dann in $R \setminus (R \setminus S)$, wenn $t \in R$ und $\not\exists (t \in R \setminus S)$ gilt. Dies ist äquivalent zu $t \in R \wedge (t \notin R \vee t \in S)$, was wiederum äquivalent ist zu $t \in R$ und $t \in S$, also $t \in R \cap S$. Damit kann der Schnittoperator durch die Differenz ausgedrückt werden.

Für die Selektion muss gezeigt werden, dass die Konjunktion, die Disjunktion und die Negation ersetzbar sind. Ein Tupel t ist genau dann in $R[\neg vga]$, wenn $t \in R$ ist und $t \not\models$ vga gilt. Dies ist äquivalent zu $t \in R$ und $t \notin R[\mathrm{vga}]$ und somit zu $t \in R \setminus R[\mathrm{vga}]$. Für Vergleichsausdrücke vga_1 und vga_2 gilt, dass t genau dann in $R[\mathrm{vga}_1 \wedge \mathrm{vga}_2]$ ist, wenn gilt $t \in R$ und $t \models \mathrm{vga}_1, \mathrm{vga}_2$. Dies ist äquivalent zu $t \in R[\mathrm{vga}_1]$ und $t \in R[\mathrm{vga}_2]$ und somit zu $t \in R[\mathrm{vga}_1 \cap \mathrm{vga}_2]$. Analog gilt $t \in R[\mathrm{vga}_1 \vee \mathrm{vga}_2]$ genau dann, wenn $t \in R[\mathrm{vga}_1] \cup R[\mathrm{vga}_2]$ gilt.

Für die Division sei $\delta = \beta \setminus \gamma$. Ein Tupel t ist genau dann in

$$R[\delta] \setminus ((R[\delta] \times S) \setminus R)[\delta],$$

wenn $t \in R[\delta]$ und nicht $t \in ((R[\delta] \times S) \setminus R)[\delta]$ gilt. Dies ist äquivalent dazu, dass ein $r \in R$ existiert mit

$$t = r|_\delta \quad \text{und} \quad t \notin ((R[\delta] \times S) \setminus R)[\delta].$$

Dies ist wiederum gleichwertig zu

$$\exists r \in R : t = r|_\delta \quad \text{und} \quad \forall r' \in ((R[\delta] \times S) \setminus R) : t \neq r'|_\delta.$$

Für den zweiten Teil kann die äquivalente Umformung

$$\exists r \in R : t = r|_\delta \quad \text{und} \quad \forall r' \in (R[\delta] \times S) : r' \notin R \wedge t \neq r'|_\delta$$

vorgenommen werden, was wiederum äquivalent ist zu

$$\exists r \in R : t = r|_\delta \quad \text{und} \quad \forall r' \in T_{\delta \cup \gamma} : r'|_\delta \in R[\delta] \wedge r'_\gamma \in S \wedge r' \notin R \wedge t \neq r'|_\delta.$$

Eine äquivalente Umformung liefert

$$\exists r \in R : t = r|_\delta \quad \text{und} \quad \forall r' \in T_\gamma : r'_\gamma \in S \wedge t \neq r'|_\delta,$$

was äquivalent ist zu $R \div S$.

Betrachte für das Join R über $\beta = \{B_1, \dots, B_{k_1}\}$ und S über $\gamma = \{C_1, \dots, C_{k_2}\}$ für $k_1, k_2 \leq |\alpha|$. Sei $\beta \cap \gamma = \{B_{i_1}, \dots, B_{i_\ell}\} = \{C_{j_1}, \dots, C_{j_\ell}\}$ mit $i_m \in [k_1], j_m \in [k_2]$ und $m \leq \min\{k_1, k_2\}$. Sei weiter

$$\gamma \cap \{C'_1, \dots, C'_{k_2}\} = \emptyset = \beta \cap \{C'_1, \dots, C'_{k_2}\}.$$

Sei weiter

$$\gamma \backslash (\beta \cap \gamma) = \{D_1, \dots, D_{k_2 - j_\ell}\}.$$

Ein Tupel t ist genau dann in

$$((R \times S[C_1 \to C'_1, \dots, C_{k_2} \to C'_{k_2}])[\bigwedge_{n \in [\ell]} B_{i_n} = C'_{j_n}])$$
$$[B_1, \dots, B_{k_1}, D_1, \dots, D_{k_2 - j_\ell}],$$

wenn ein $t \in ((R \times S[C_1 \to C'_1, \dots, C_{k_2} \to C'_{k_2}])[\bigwedge_{n \in [\ell]} B_{i_n} = C'_{j_n}]$ existiert mit $t' = t|_{B_1, \dots, B_{k_1}, D_1, \dots, D_{k_2 - j_\ell}}$. Dies ist äquivalent zu der Existenz ein $t' \in R$ und der Existenz einen $t'' \in S$ mit

$$t''(C_i) = t'(C'_i) \wedge t' = t|_{B_1, \dots, B_{k_1}, D_1, \dots, D_{k_2 - j_\ell}},$$

was wiederum äquivalent ist zur Existenz von $t' \in R$ und $t'' \in S$ mit $t|_\beta = t'$ und $t|_\gamma = t''$.

Da sich jede Relation als direktes Produkt von Relationen der Form $\{(A : c)\}$ und Vereinigung darstellen lässt, ist die Behauptung bewiesen. $\square$

Da in der Behauptung das Komplement $\bar{\cdot}$ weder explizit genannt noch ausgeschlossen wurde, möchten wir noch anmerken, dass wenn das Komplement in e zugelassen ist, in e' wegen $R \backslash S = \overline{\overline{R} \cup S}$, die Differenz nicht benötigt wird, dafür aber das Komplement aufgenommen werden muss.