

## 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  $(L, \mu)$  definiert ist und die Relationenalgebra  $(L', \mu')$  ist. Hierzu gilt zu zeigen, dass dies für ein beliebiges Datenbankschema  $\sigma$  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 einzeln 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  $e_{Rel} = e'_{Rel}[es_1, \dots, es_n]$  mit  $n > 0$  und  $es_i$  Vergleichsausdrücke mit  $0 < i \leq n$ . Dann lässt sich  $e$  in Datalog als  $e''_{Dat}(V) : -e'_{Dat}(V), es_1(V), \dots, es_n(V)$  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  $e_{Rel}[k \rightarrow j]$ , so gilt  $k \in V \wedge k \notin V' \wedge j \notin V \wedge j \in V'$ . 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  $x_1, \dots, x_n, y_1, \dots, y_m$  mit  $n = |Param_{e_{Rel}}|$ ,  $m = |Param_{e'_{Rel}}|$ ,  $x_i \in Param_{e_{Rel}}$  für  $1 \leq i \leq n$ ,  $y_i \in Param_{e'_{Rel}}$  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  $e''_{Rel} = e_{Rel} \setminus e'_{Rel}$  gegeben. Sei der Parameter  $Param_{e_{Rel}}$  definiert als alle Spalten im Resultat von  $e_{Rel}$ ,<sup>1</sup> so lässt sich die Differenz definieren als Regel

$$e''_{Dat}(Param_e) :- e_{Dat}(Param_e), \text{ not } e'_{Dat}(Param_e).$$

**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 äquivalente 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  $e_{Rel}$  und  $e'_{Rel}$  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.

---

<sup>1</sup>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 dank Closed World Assumption umschreiben als  $(Tab_e \setminus e)[V]$ , 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 Prolog ist

$a(A,B) :- R(A,B, \_).$

$a(A,B) :- S(A,D), T(\_,D,B).$

ad. b)

Die äquivalente Darstellung in Prolog ist

$sbt(A,B,C,G) :- S(A,C), T(B,C,G).$

$r'(A,B,C) :- R(A,B,C), A > B.$

$r'(A,B,C) :- R(A,B,C), C > 5.$

$rbs(A,B,C,G) :- r'(A,B,C), S(A,G).$

$q(A,B,C,G) :- sbt(A,B,C,G), \text{ not } rbs(A,B,C,G).$

### Aufgabe 3.