

Gruppe 22/48

Tutorübung zu Einsatz und Realisierung von Datenbanksystemen (SS 18)

Michael Schwarz

Institut für Informatik
Technische Universität München

08.05. + 15.05 / 17.05.2018

Übung 22

- Besprechung (fast) aller Aufgaben von Blatt 4 (auch solche, die mit "wird nicht besprochen" gekennzeichnet sind)
- **Am 08.05 und 15.05 (je Termin +1 Anwesenheitspunkt)**

Übung 48

- Besprechung einiger Aufgaben von Blatt 4
- **Am 17.05**
- Wer in einer Donnerstagsgruppe angemeldet ist, braucht keine Ersatzübung zu besuchen (für 10.05 automatisch +1)

Naive Auswertung durch Iteration

$$A(V, N) = V_S(V, N) \cup \Pi_{V, N}(A(V, M) \bowtie V_S(M, N))$$

A := {}: /*Initialisierung auf die leere Menge */

repeat

 A' := A;

 A := V_S(V, N); /* erste Regel */

 A := A \cup $\Pi_{V, N}(A'(V, M) \bowtie V_S(M, N))$; /* zweite Regel */

until A' = A

output A;

Programm zur semi-naiven Auswertung von *aufbauen*

1. $A := \{ \}; \Delta V_S := \{ \};$
2. $\Delta A := V_S(V, N);$ /* erste Regel */
3. $\Delta A := \Delta A \cup \Pi_{V, N}(A(V; M) \bowtie V_S(M, N));$ /* zweite Regel */
4. $A := \Delta A;$
5. **repeat**
6. $\Delta A' := \Delta A;$
7. $\Delta A := \Delta V_S(V, N);$ /* erste Regel, liefert \emptyset */
8. $\Delta A := \Delta A \cup$ /* zweite Regel */
9. $\Pi_{V, N}(\Delta A'(V, M) \bowtie V_S(M, N)) \cup$
10. $\Pi_{V, N}(A(V, M) \bowtie \Delta V_S(M, N));$
11. $\Delta A := \Delta A - A;$ /* entferne "neue" Tupel, die schon vorhanden waren */
12. $A := A \cup \Delta A;$
13. **until** $\Delta A = \emptyset;$

Negation im Regelrumpf

indirektAufbauen(V, M) :- aufbauen(V, M), \neg voraussetzen(V, M)

Stratifizierte Datalog-Programme

Eine Regel mit einem negierten Prädikat im Rumpf, wie z.B.

$$r \equiv p(\dots) :- q_1(\dots), \dots, \neg q_i(\dots), \dots, q_n(\dots).$$

kann nur dann sinnvoll ausgewertet werden, wenn Q_i schon vollständig materialisiert ist. Also müssen zuerst alle Regeln mit Kopf $q_i(\dots) :- \dots$ ausgewertet sein. Das geht nur, wenn q_i nicht abhängig von p ist.

Also darf der Abhängigkeitsgraph keine Pfade von q_i nach p enthalten. Wenn das für alle Regeln der Fall ist, nennt man das Datalog-Programm *stratifiziert*.

Sicherheit von Datalog-Regeln

- Es gibt unsichere Regeln, wie z.B.

$$\text{ungleich}(X, Y) \text{ :- } X \neq Y.$$

Diese definieren unendliche Relationen.

- Eine Datalog-Regel ist sicher, wenn alle Variablen im Kopf beschränkt (range restricted) sind. Dies ist für eine Variable X dann der Fall, wenn:
 - ☀ die Variable im Rumpf der Regel in mindestens einem normalen Prädikat – also nicht nur in eingebauten Vergleichsprädikaten – vorkommt oder
 - ☀ ein Prädikat der Form $X = c$ mit einer Konstante c im Rumpf der Regel existiert oder
 - ☀ ein Prädikat der Form $X = Y$ im Rumpf vorkommt, und man schon nachgewiesen hat, dass Y eingeschränkt ist.