



## Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken* im SoSe20

Maximilian {Bandle, Schüle}, Josef Schmeißer (i3erdb@in.tum.de)

<http://db.in.tum.de/teaching/ss20/impldb/>

### Blatt Nr. 05

**Hinweise** Die Datalogaufgaben können auf <https://datalog.db.in.tum.de/> getestet werden. Auf der Seite kann unter *examples* ein entsprechender Datensatz geladen werden. Die neuen IDB Regeln sollten am Ende der EDB definiert und dann im Query-Eingabefeld abgefragt werden.

Zusätzlich zu der in der Vorlesung vorgestellten Syntax hier noch eine Kurzübersicht der Vergleichsoperatoren:  $X < Y, Y > X$  (kleiner, größer),  $X = < Y, X >= Y$  (kleiner oder gleich, größer oder gleich),  $X = Y, X \neq Y$  (gleich, ungleich),  $\text{not}(\text{pred}(X, Y))$  (existiert nicht  $\text{pred}(X, Y)$ ).

### Hausaufgabe 1

Definieren Sie das Prädikat  $\text{sg}(X, Y)$  das für “same generation” steht. Zwei Personen gehören zur selben Generation, wenn Sie mindestens je ein Elternteil haben, das derselben Generation angehört.

Verwenden Sie beispielsweise die folgende Ausprägung einer ElternKind Relation. Das erste Element ist hier das Kind, das Zweite ein Elternteil.

```
parent(c, a).  
parent(d, a).  
parent(d, b).  
parent(e, b).  
parent(f, c).  
parent(g, c).  
parent(h, d).  
parent(i, d).  
parent(i, e).  
parent(f, e).  
parent(j, f).  
parent(j, h).  
parent(k, g).  
parent(k, i).
```

- Definieren Sie das Prädikat in Datalog.
- Demonstrieren Sie die naive Ausführung des Prädikats.
- Erläutern Sie das Vorgehen bei der seminaiven Auswertung.

### Gruppenaufgabe 2

Ist folgendes Datalog-Programm stratifiziert?

$$\begin{aligned} p(X, Y) & :- q_1(Y, Z), \neg q_2(Z, X), q_3(X, P). \\ q_2(Z, X) & :- q_4(Z, Y), q_3(Y, X). \\ q_4(Z, Y) & :- p(Z, X), q_3(X, Y). \end{aligned}$$

Ist das Programm sicher – unter der Annahme, dass  $p, q_1, q_2, q_3, q_4$  IDB- oder EDB-Prädikate sind?

### Hausaufgabe 3

Gegeben seien die Tabellen **Studenten** und **Punkte** mit Schlüssel **MatrNr**, wobei **Punkte** auf einem separaten Rechner gespeichert ist. Es soll folgende Anfrage ausgeführt werden:

```
SELECT Name, Bonus FROM Student s, Punkte p WHERE s.MatrNr = p.MatrNr;
```

Der Datenbankadministrator entscheidet sich für einen Bloom-Filter zur Vorauswahl der Tupel. Auf **MatrNr** wird die Hash-Funktion  $h(x) = x \bmod 5$  angewendet.

Studenten			Punkte		
MatrNr	Name	Hashwert	MatrNr	Bonus	Hashwert
27	Magda		27	ja	
4	Josef		16	nein	
19	Erik		25	nein	
95	Philipp		95	ja	

- a) Berechnen Sie die Hash-Werte und tragen Sie diese in die obige Tabelle ein.
- b) Geben Sie den von **Studenten** zu übertragenden Bitvektor an.
- c) Geben Sie basierend auf dem Bitvektor an, welche Tupel aus **Punkte** übertragen werden.
- d) Geben Sie die Falsch-Positiv-Rate (false positive rate) an.
- e) Nehmen Sie an, dass jedes Tupel 8 Byte und der Bloomfilter selbst 1 Byte groß ist. Berechnen Sie zunächst die übertragenen Bytes ohne und mit Einsatz des Bloom-Filters.

### Hausaufgabe 4

Überlegen Sie sich, welche Tupel bei der Anwendung des bloomfilterbasierten Joins in Abbildung 1 übertragen werden. Markieren Sie insbesondere, welche Tupel übertragen werden, obwohl sie keinen Joinpartner finden (sog. *false drops*). Wie kann die Anzahl dieser *false drops* verringert werden? Welche Eigenschaften sollte die Hashfunktion  $h(c)$  die bei dieser Joinbearbeitung verwendet wird erfüllen?

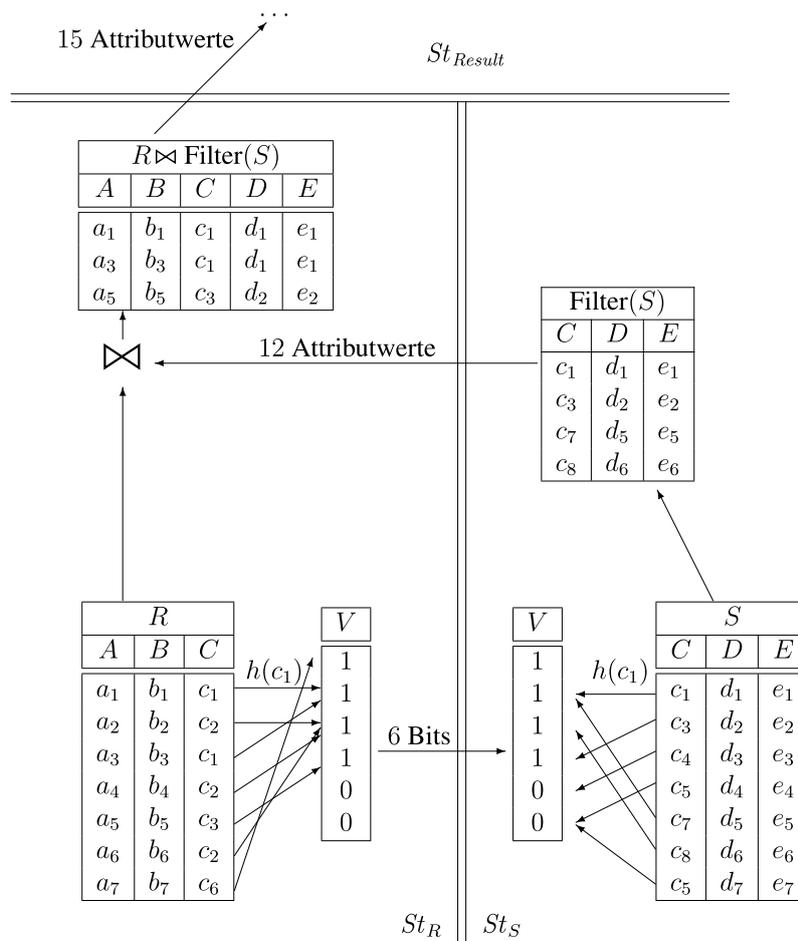


Abbildung 1: Beispiel einer verteilten Joinbearbeitung mit Bloomfilter.