



Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken* im SoSe25

Alice Rey, Maximilian Reif, Tobias Goetz (i3erdb@in.tum.de)

<http://db.in.tum.de/teaching/ss25/impldb/>

Blatt Nr. 05

Hinweise Die Datalogaufgaben können auf <https://souffle.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), $X \leq Y, X \geq Y$ (kleiner oder gleich, größer oder gleich), $X = Y, X \neq Y$ (gleich, ungleich), $\text{!pred}(X, Y)$ (existiert nicht $\text{pred}(X, Y)$).

Hausaufgabe 1

Bleiben wir bei dem bekannten Universitätsschema:

```
Assistenten(PersNr, Name, Fachgebiet, Boss)
 hoeren(MatrnNr, VorlNr)
 pruefen(MatrnNr, VorlNr, PersNr, Note)
 Vorlesungen(VorlNr, Titel, SWS, gelesenVon)
 Professoren(PersNr, Name, Rang, Raum)
 voraussetzen(Vorg, Nachf)
 Studenten(MatrnNr, Name, Semester)
```

Formulieren Sie folgende Anfragen in Datalog und testen Sie sie:

- a) Geben Sie alle *Professoren* an, die mindestens eine Prüfung abgehalten haben.

```
.decl pruefendeProfs(name: symbol)
```

- b) Übersetzen Sie folgenden Ausdruck des Domänenkalküls in Datalog. Machen Sie sich der Bedeutung des Ausdrucks bewusst.

$$\{[t] \mid \exists v, s, g([v, t, s, g] \in \text{Vorlesungen} \wedge \exists v2([v, v2] \in \text{voraussetzen} \wedge \exists s2, g2([v2, \text{'Wissenschaftstheorie', s2, g2}] \in \text{Vorlesungen})))\}$$

- c) Joinen Sie die nachfolgende Datalog-Anfrage so, dass die Titel der Vorlesungen ausgegeben werden. Was bedeutet diese Anfrage?

```
.decl geschwisterVL(N1: number, N2: number)
 .decl nahverwandtVL(N1: number, N2: number)
 geschwisterVL(N1, N2) :- voraussetzen(V, N1), voraussetzen(V, N2), N1 < N2.
 nahverwandtVL(N1, N2) :- geschwisterVL(N1, N2).
 nahverwandtVL(N1, N2) :- geschwisterVL(M1, M2), voraussetzen(M1, N1),
                          voraussetzen(M2, N2).
```

Hausaufgabe 2

Geben Sie Datalog Regeln an, die Studenten (Namen angeben) finden, die von einem Prüfer geprüft worden, der selbst nicht die geprüfte Vorlesung gehalten hat. Das korrekte Ergebnis für diese Anfrage ist Russels Prüfling, Carnap. Führen Sie die Anfrage im Datalog Tool aus! `.decl fremdgeprueft(SN: symbol, PID: number, VPID: number)`

Hausaufgabe 3

Definieren Sie das Prädikat `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.

```
.decl parent(child: symbol, parent: symbol)
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").
```

a) Definieren Sie das Prädikat in Datalog.

```
.decl sg(h1: symbol, h2: symbol)
```

b) Demonstrieren Sie die naive Ausführung des Prädikats.

c) Erläutern Sie das Vorgehen bei der seminaiven Auswertung.

Gruppenaufgabe 4

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?

Gruppenaufgabe 5

Gegeben sei folgende Faktenbasis, die einen direkten azyklischen Graphen (DAG) darstellt.

```
.decl kante(a: number, b: number)
kante(1,2).
kante(2,3).
kante(3,4).
kante(2,5).
kante(5,3).
```

1. Geben Sie in Datalog ein Prädikat $\text{pfad}(V,N,L)$ an, dass alle möglichen Pfade von V nach N mit Länge L ausgibt.

```
.decl pfad(von: number, nach: number, laenge: number)
```

2. Geben Sie nun das Prädikat $\text{kuerzestePfade}(V,N,L)$ an, das pro Beginn V und Ziel N nur den kürzesten Pfad ausgibt.
3. Bestimmen Sie nun den längsten kürzesten Pfad `.decl laengsterkuerzesterPfad(L: number)`.
4. Erstellen Sie in SQL eine rekursive CTE $\text{pfad}(V,N,L)$, die die Länge aller Pfade im DAG ausgibt.
5. Basierend auf $\text{pfad}(V,N,L)$, geben Sie die Länge des längsten kürzesten Pfades aus.

Hausaufgabe (wird nicht in der Übung besprochen)

Schreiben Sie zu dem U-Bahn-Netz-Beispiel auf der Datalog Seite (unter Examples) folgende Anfragen in Datalog:

1. Erstellen Sie den Stationsplan für den U-Bahnhof Fröttmanning, der alle Stationen, die ohne Umstieg erreichbar sind, auflistet.
2. Erstellen Sie für Garching-Forschungszentrum einen Plan, der alle erreichbaren Stationen, die minimale Anzahl an Umstiegen und Stops auflistet. Beschreiben Sie Ihren Ansatz ausführlich.
3. Gibt es zwischen allen Paaren von Stationen Verbindungen mit maximal einmaligem Umsteigen? Versuchen Sie Gegenbeispiele mittels Datalog zu finden.