



Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken* im SoSe21

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

<http://db.in.tum.de/teaching/ss21/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), $not(pred(X, Y))$ (existiert nicht $pred(X, Y)$).

Hausaufgabe 1

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.

```
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.

```
sg(X, Y) = :- parent(Z, X), X=Y. % X als Elternteil
sg(X, Y) = :- parent(X, Z), X=Y. % X als Kind
% X, Y Kind von U und V, U und V gleiche Generation
sg(X, Y) = :- sg(U, V), parent(X, U), parent(Y, V).
```

b) Demonstrieren Sie die naive Ausführung des Prädikats.

c) Erläutern Sie das Vorgehen bei der seminaiven Auswertung.

Siehe Übungsbuch

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?

Loesung: Vgl. Übungsbuch. Das Programm ist **nicht stratifiziert**, aber **sicher**. Es ist nicht stratifiziert, weil q_2 von p abhängt, aber negiert in p vorkommt. Es ist sicher, weil alle Variablen in IDB- oder EDB-Prädikaten gebunden sind.

Zur Wiederholung: Eine Regel ist **sicher** gdw. alle Variablen **eingeschränkt** sind. Variable X ist in einer Regel **eingeschränkt**, falls sie im Rumpf enthalten ist und sie:

- in einem positiven Prädikat vorkommt (nicht Vergleichsprädikat),
- $X = c$ (Konstante) oder
- $X = Y$, wenn Y bereits nachgewiesen ist.

Jede Variable eines negierten Prädikates muss bereits eingeschränkt sein.^{ab}

Ein Datalog-Programm ist **stratifiziert**, wenn in einer Regel p alle negierten Prädikate $\text{not}(q_i)$ nicht von p abhängen (kein Zyklus) und alle positiven Prädikate vorher oder unabhängig von der Reihenfolge (wie bei Rekursion) ausgewertet werden. Formal ausgedrückt können wir jedem Prädikat eine sogenannte Stratifikationsnummer zuordnen. Negierte Prädikate müssen eine echt kleinere Stratifikationsnummer besitzen, positive Prädikate eine maximal so große Stratifikationsnummer wie das davon abhängige Prädikat.

^a<https://www.dbis.informatik.uni-goettingen.de/Teaching/DB/db-datalog.pdf>

^b<http://pages.cs.wisc.edu/~paris/cs784-s17/lectures/lecture9.pdf>

Hausaufgabe 3

U-Bahnen sind toll! Nehmen wir als Faktenbasis die U-Bahnstationen der Linie U2 Richtung Messestadt West und der Linie U6 Richtung Klinikum Großhadern.

`direkt(Von, Ziel, Linie).`

Formulieren Sie ein Datalog-Prädikat, das Ihnen von Garching-Forschungszentrum aus kommend die erreichbaren Stationen inklusiver der Anzahl der Stationen angibt. Testen Sie es!

`indirekt(A,C,S) :- direkt(A,C,_),S=0.`
`indirekt(A,C,S) :- indirekt(A,B,R),direkt(B,C,_),S=R+1.`
`indirekt(garching_forschungszentrum,C,S).`

Hausaufgabe 4

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.

```
bidirekt(A,B,L) :- direkt(A,B,L), A\=B.  
bidirekt(A,B,L) :- direkt(B,A,L), A\=B.  
bidirekt(A,B,L) :- bidirekt(A,X,L), direkt(X,B,L), A\=B.  
  
bidirekt(froettmanning,B,_)
```

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.

Vereinfachte Lösung (betrachten nur in Fahrtrichtung)

```
% Erreichbar naechster Stop
aufwand(A,B,L,S,U) :- direkt(A,B,L), S=0, U=0.
% Erreichbar auf gleicher Linie
aufwand(A,B,L,S,U) :- aufwand(A,C,L,SX,UX), direkt(C,B,L),
    S=SX+1, U=UX.
% Erreichbar durch umsteigen
aufwand(A,B,L,S,U) :- aufwand(A,C,LA,SX,UX), direkt(C,B,LB),
    S=SX+1, LA\=LB, L=LB, U=UX+1.
```

Lösung mit Richtungs- und Linienwechsel.

```
% Merke Richtung in die gefahren wird (R=vorwaerts oder rueckwaerts)
bdirekt(A,B,L,R) :- direkt(A,B,L), R=v.
bdirekt(A,B,L,R) :- direkt(B,A,L), R=r.
```

```
% Maximale Anzahl der Stops, ist noetig falls die Rekursion
% in einem Kreis im Graph festhaengt.
% Ohne Aggregation einfach 49 statt SMAX bei aufwand(...) einsetzen
smax(MAX):- count(direkt(_,_,_), MAX).
```

```
% Erreichbar naechster Stop
aufwand(A,B,L,R,S,U) :- bdirekt(A,B,L,R),
    S=1, U=0, A\=B.
% Erreichbar auf gleicher Linie
aufwand(A,B,L,R,S,U) :- aufwand(A,C,L,R,SX,UX), bdirekt(C,B,L,R),
    S=SX+1, U=UX, A\=B, smax(SMAX), S<SMAX.
% Erreichbar durch Umsteigen auf andere Linies.
% Richtungswechsel erlaubt.
aufwand(A,B,L,R,S,U) :- aufwand(A,C,LA,_,SX,UX), bdirekt(C,B,LB,R),
    S=SX+1, LA\=LB, L=LB, U=UX+1, A\=B, smax(SMAX), S<SMAX.
```

Im *aufwand* Prädikat ist ein Richtungswechsel ohne gleichzeitigen Linienwechsel nicht berücksichtigt. Dies ist auch nicht notwendig, da am Startpunkt durch *bdirekt* in beide Richtungen gestartet werden kann, und bei einem Linienwechsel dann auch jedesmal die Möglichkeit besteht, die Richtung frei zu wählen.

Wegen der Struktur der Linien im Beispielgraph (sie treffen sich nur am Sendlinger Tor) ist die Lösung des *aufwand* Prädikats schon jeweils die kürzeste Strecke. Bei einem Netz mit zwei Treffpunkten wären durch die Richtungswechsel Kreise möglich und das Minimum für jede Strecke müsste mit folgendem Prädikat gefunden werden.

```
minaufwand(A,B,S,U) :- aufwand(A,B,_,_,S,U),
    min(aufwand(A,B,_,_,ST,U), ST,S),
    min(aufwand(A,B,_,_,S,UM),UM,U).
```

```
minaufwand(garching_forschungszentrum,B,S,U)
```