



Übung zur Vorlesung

Einsatz und Realisierung von Datenbanksystemen im SoSe14

Moritz Kaufmann (moritz.kaufmann@tum.de)

<http://www-db.in.tum.de/teaching/ss14/impldb/>

Blatt Nr. 4

Bitte führen Sie alle anzugebenden Datalog-Anfragen tatsächlich aus!

Beispielsweise hier: <http://datalog.db.in.tum.de>

Aufgabe 1

Gegeben sei die nachfolgende *KindEltern*-Ausprägung für den Stammbaum-Ausschnitt der griechischen Götter und Helden:

KindEltern		
Vater	Mutter	Kind
Zeus	Leto	Apollon
Zeus	Leto	Artemis
Kronos	Rheia	Hades
Zeus	Maia	Hermes
Koios	Phoebe	Leto
Atlas	Pleione	Maia
Kronos	Rheia	Poseidon
Kronos	Rheia	Zeus

```
ke(zeus, leto, apollon).
ke(zeus, leto, artemis).
ke(kronos, rheia, hades).
ke(zeus, maia, hermes).
ke(koios, phoebe, leto).
ke(atlas, pleione, maia).
ke(kronos, rheia, poseidon).
ke(kronos, rheia, zeus).
```

Formulieren Sie folgende Anfragen in Datalog:

a) Bestimmen Sie alle Geschwisterpaare.

```
parent(P,K) :- kindEltern(P,_,K).
parent(P,K) :- kindEltern(_,P,K).

sibling(A,B) :- parent(P,A), parent(P,B), A\=B.
```

b) Ermitteln Sie Paare von Cousins und Cousinen beliebigen Grades. Die Definition finden Sie auf Wikipedia.

```
cousin(A,B) :- parent(PA,A), parent(PB,B), sibling(PA,PB).
cousin(A,B) :- parent(PA,A), parent(PB,B), cousin(PA,PB).
```

- c) Geben Sie alle Verwandtschaftspaare. Überlegen Sie sich eine geeignete Definition von Verwandtschaft und setzen Sie diese in Datalog um.

```
related(A,B) :- sibling(A,B).
related(A,B) :- related(A,C),parent(C,B).
related(A,B) :- related(C,B),parent(C,A).
```

- d) Bestimmen Sie alle Nachfahren von Kronos. Formulieren Sie die Anfrage auch in SQL, so dass sie unter PostgreSQL ausführbar ist (online testen unter: <http://hyper-db.com/interface.html>). Sie können die Daten als Common Table Expression definieren und dann nutzen:

```
WITH
kindEltern(vater,mutter,kind) as (
  VALUES
    ('Zeus', 'Leto', 'Apollon'),
    ('Zeus', 'Leto', 'Artemis'),
    ('Kronos', 'Rheia', 'Hades'),
    ('Zeus', 'Maia', 'Hermes'),
    ('Koios', 'Phoebe', 'Leto'),
    ('Atlas', 'Pleione', 'Maia'),
    ('Kronos', 'Rheia', 'Poseidon'),
    ('Kronos', 'Rheia', 'Zeus')
),
parent(eltern,kind) as (
  select vater, kind from kindEltern UNION
  select mutter, kind from kindEltern
)
select * from parent where eltern='Zeus'
```

Datalog

```
nachfahr(P,N) :- parent(P,N).
nachfahr(P,N) :- nachfahr(P,X),nachfahr(X,N).
```

Alternativ

```
nachfahr(P,N) :- parent(P,N).
nachfahr(P,N) :- nachfahr(P,X),parent(X,N).
```

Anfrage für die Nachfahren von Kronos

```
nachfahr(kronos,X).
```

SQL

```
WITH
kindEltern(vater,mutter,kind) as (
VALUES
('Zeus', 'Leto', 'Apollon'),
('Zeus', 'Leto', 'Artemis'),
('Kronos', 'Rheia', 'Hades'),
('Zeus', 'Maia', 'Hermes'),
('Koios', 'Phoebe', 'Leto'),
('Atlas', 'Pleione', 'Maia'),
('Kronos', 'Rheia', 'Poseidon'),
('Kronos', 'Rheia', 'Zeus')
),
parent(eltern, kind) as(
select vater, kind from kindEltern UNION
select mutter, kind from kindEltern
),
RECURSIVE nachfahren(person, nachfahre) AS (
SELECT * from parent
UNION ALL
SELECT n.person, p.kind FROM nachfahren n, parent p
WHERE p.eltern = n.nachfahre
)
select * from nachfahren WHERE person='Kronos'
```

Aufgabe 2

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

1. Erstellen Sie den Stationsplan für den U-Bahnhof Fröttmanning, der alle Station, die ohne umsteigen 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* Praedikat ist ein Richtungswechsel ohne gleichzeitigen Linienwechsel nicht beruecksichtigt. Dies ist auch nicht notwendig, da am Startpunkt durch *bdirekt* in beide Richtungen gestartet werden kann, und bei einem Linienwechsel dann auch jedesmal die Moeglichkeit besteht die Richtung frei zu waehlen.

Wegen Struktur der Linien im Beispielgraph (sie treffen sich nur am Sendlinger Tor) ist die Loesung des *aufwand* Praedikats schon jeweils die kuerzeste Strecke. Bei einem Netz mit zwei Treffpunkten waeren durch die Richtungswechsel Kreise moeglich und das Minimum fuer jede Strecke muesste mit folgendem Praedikat 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)
```

Aufgabe 5

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?

Siehe Übungsbuch