

f:\mk-lehre\21fsw-se3\g0-do10-12se3g0-g0-a02.off.pl

Page 1

```

*****
% Aufgabenblatt 02 - SE3-IP WiSe 16/17
%
% Finn-Lasse Jörgensen 6700628 4joergen@informatik.uni-hamburg.de
% Fabian Behrendt 6534523 fabian.behrendt95@gmail.com
% Daniel Klotzsche 6535732 daniel.klotzsche@hotmail.de
%
% Wir sind bereit folgende Aufgaben zu präsentieren:A1,A3,A2.1,A2.2,A4
*****
[haeuser]

```

A1: 2,5P

-0,5P

```

%% A1 %%
/*
mutter von(P1,X),vater von(X,P2).
P1 ist die Oma von P2 %väterlicherseits
Frage2?
mutter von(X,P1),mutter von(Y,X),mutter von(Y,P2),X=P2.
P2 ist die/der Tante/Onkel von P1

vater von(X,P1),mutter von(Y,X),mutter von(Y,Z),mutter von(Z,P2),X=Z.
P1 ist der/die Cosain/Cosine von P2

mutter von(X,P1),mutter von(Y,P2),vater von(Z,P1),vater von(Z,P2),P1=P2,!=Y.
P1 und P2 sind halbgeschwister (Gleicher Vater unterschiedliche Mutter)
*/

%% A2 %%
%% 1.
*****
?- obj(,efh,Strasse,).
% -> Strasse = gaertnerstr ;
% -> Strasse = bahnhofsstr ;
% -> Strasse = bahnhofsstr ;
% -> Strasse = gaertnerstr.

%% 2.
?- obj(Nr,Typ,Strasse,Hausnumm, Baujahr), Baujahr > 2006.
% -> false.
% Es ist kein Haus in der Datenbank, welches in den letzten 10 Jahren gebaut wur
rde.
?- obj(Nr,Typ,Strasse,Hausnumm, Baujahr), Baujahr > 2006.
% -> Nr = 8,
% -> Typ = informatikum,
% -> Strasse = vogt koelln strasse,
% -> Hausnummer = 30,
% -> Baujahr = 2010.

%% 3.
?- obj(Nr, ,Strasse,HsNr, ),\+obj(Nr,efh,Strasse,HsNr, ). % einfacher: Typ != efh
% -> Nr = 4,
% -> Strasse = bahnhofsstr,
% -> HsNr = 28 ;
% -> Nr = 5,
% -> Strasse = bahnhofsstr,
% -> HsNr = 30 ;
% -> Nr = 6,
% -> Strasse = bahnhofsstr,
% -> HsNr = 26 ;
% -> Nr = 8,
% -> Strasse = vogt koelln strasse,
% -> HsNr = 30.

%% 4.
?- bew(,Nr1, ,K1, ,Datum),Datum>20111103, bew(,Nr2,V2, , ,),Nr1=Nr2,K1=V2.

```

f:\mk-lehre\21fsw-se3\g0-do10-12se3g0-g0-a02.off.pl

Page 2

```

% -> false.
% Keine passenden Einträge.
?- assert(bew(5, ,stadthamburg, danielklotzsche, 1000000, 2012110
?- assert(bew(5, 8, danielklotzsche, donaldrump, 10000000, 2014010

?- bew(,Nr1, ,K1, ,Datum),Datum>20111103, bew(,Nr2,V2, , ,),Nr1=Nr2,K1=V2.
% -> Nr1 = Nr2, Nr2 = 8,
% -> K1 = V2, V2 = danielklotzsche,
% -> Datum = 20121103 ;
% -> false.

%% 5.
*****
?- obj(Nr, ,gaertnerstr, ,),bew(,Nr,Verkaeuf, Kaeuf,Preis1, ),(bew(, , , ,
,PreisGroesser, ),PreisGroesser>Preis1, %das liefert nur EIN TEURERES Haus

%% A3 %%
/*
:- dynamic anschluss/2.

anschluss(NR,Name)
anschluss(0,otter).
anschluss(1,hans).
anschluss(2,peter).
anschluss(3,hellen).
anschluss(4,petra).
anschluss(5,lars).

:- dynamic ip/4.

%ip(ip,von,bis,anschlussNR)
ip("0.0.0.1",20120101,20130101,0).
ip("0.2.0.0",20120101,20120307,1).
ip("0.0.1.0",20120307,20130101,1).
ip("10.0.0.1",20120101,20130101,4).
ip("0.10.0.1",20120101,20121120,3).
ip("0.10.3.1",20121120,20130101,3).
ip("0.10.66.1",20120301,20120701,2).
ip("0.0.1.0",20130101,20131001,3).

:- dynamic connection/3.

%connection(ip,ip,zeitpunkt)
connection("0.0.0.1",0.2.0.0,20120203).
connection("0.0.0.1",0.10.66.1,20120203).
connection("0.10.0.1",10.0.0.1,20120403).

%Abstraktion?
%%3.
%Welche IP wurde wem zugewiesen
?- anschluss(ID,Name), ip(IP, , ,ID).
%Wer hatte wann mit jemanden eine connection
?- connection(A,B,T), ip(A,AT1,AT2,IDA), AT1=AT2, T=AT2, ip(B,BT1,BT2,IDB)
BT1<T, T=BT2, anschluss(IDA,AName), anschluss(IDB,BName).
%Wann wurde eine IP benutzt
?- ip("0.0.1.0",Von,Bis, ).
%Wer hatte eine connection zu sich selber
?- connection(IP,IP,T), ip(IP,T1,T2,ID), T1<T, T2=T, anschluss(ID,Name).
%Alle registrierten Benutzer die eine IP hatten
?- anschluss(ID,Name), ip(, , ,ID).

/*
%% A4 %%
Fakt: Fakten sind elementare Klauseln einer Datenbank, welche jeweils ein Attribut
nt
einer Relation mit den angegebenen Prädikatnamen und den dazugehörigen
Argumenten spezifizieren, wobei eine unterschiedliche Stelligkeit desselben
Prädikatnamens eine unterschiedliche Relation definiert.

```

-0,5P

A3: 5P

A2: 3,5P

-1P

A4: 2P

C9: 3.6.2017, 1.0.2015, 1.0.2015

Regel: Regeln sind Einträge bzw. Klauseln in einer Datenbank. Diese besagen, dass bei Gültigkeit eines bestimmten Faktus ein anderer, durch die Regel definierter Fakt ebenfalls gilt.

Mithilfe von Regeln lassen sich neue Fakten aus bisher bekannten Fakten ableiten.

Durch Regeln wird eine mögliche Redundanz vermieden und es können Relationen über unendlichen Domänen spezifiziert werden.

Anfrage: Anfragen sind ebenfalls elementare Klauseln einer Datenbank, welche am Systemprompt einzugeben sind. Sie prüfen anhand der Fakten der Datenbasis auf Konsistenz unter Berücksichtigung der closed world assumption (alle Fakten, welche nicht in der Datenbasis vorliegen, sind als falsch anzunehmen).

\*/