

1 Übungsblatt

1.1 Aufgabe

1.1.1

```
3 ?- ['C://Users/Elias/Dropbox/Studium/SE3 LP/L\"osungen/Aufgabenblatt 1/familie.pl  
true.
```

Die Hochkommas sind notwendig, da in dem Pfad Sonderzeichen vorkommen.

1.1.2

```
4 ?- listing([mutter_von,vater_von]).
```

```
:- dynamic mutter_von/2.
```

```
mutter_von(marie, hans).  
mutter_von(marie, helga).  
mutter_von(julia, otto).  
mutter_von(barbara, klaus).  
mutter_von(barbara, andrea).  
mutter_von(charlotte, barbara).  
mutter_von(charlotte, magdalena).
```

```
:- dynamic vater_von/2.
```

```
vater_von(otto, hans).  
vater_von(otto, helga).  
vater_von(gerd, otto).  
vater_von(johannes, klaus).  
vater_von(johannes, andrea).  
vater_von(walter, barbara).  
vater_von(walter, magdalena).  
true.
```

1.1.3

```
5 ?- assert(vater_von(homer,bart)).  
true.
```

```
6 ?- asserta(vater_von(homer,lisa)).  
true.
```

```
7 ?- assertz(mutter_von(marge,lisa)).  
true.
```

```
8 ?- assertz(mutter_von(marge,bart)).  
true.
```

asserta/1 fügt die Klausel an der 1. Stelle in die Datenbasis ein. assertz/1 fügt die Klausel am Ende der Datenbasis ein. assert/1 ist die veraltete Form von assertz/1.

1.2

1.2.1

a)

```
9 ?- mutter_von(julia,otto).  
true.
```

b)

```
10 ?- vater_von(otto,helga).  
true.
```

c)

```
11 ?- vater_von(X,julia).  
false.
```

d)

```
12 ?- vater_von(otto,X).  
X = hans ;  
X = helga.
```

e)

```
13 ?- vater_von(X,Y);mutter_von(X,Y).  
X = homer,  
Y = lisa ;  
X = otto,  
Y = hans ;  
X = otto,  
Y = helga ;  
X = gerd,  
Y = otto ;  
X = johannes,  
Y = klaus ;
```

```
X = johannes,  
Y = andrea ;  
X = walter,  
Y = barbara ;  
X = walter,  
Y = magdalena ;  
X = homer,  
Y = bart ;  
X = marie,  
Y = hans ;  
X = marie,  
Y = helga ;  
X = julia,  
Y = otto ;  
X = barbara,  
Y = klaus ;  
X = barbara,  
Y = andrea ;  
X = charlotte,  
Y = barbara ;  
X = charlotte,  
Y = magdalena ;  
X = marge,  
Y = lisa ;  
X = marge,  
Y = bart.
```

f)

```
14 ?- \+ vater_von(hans,X).  
true.
```

g)

```
15 ?- \+ vater_von(johannes,X).  
false.
```

h)

```
16 ?- \+(\+ vater_von(otto,X)).  
true.
```

1.2.2

```
17 ?- mutter_von(charlotte,X), (vater_von(X,Y);mutter_von(X,Y)).  
X = barbara,
```

```
Y = klaus ;
X = barbara,
Y = andrea ;
false.
```

1.2.3

```
18 ?- trace.
```

```
true.
```

```
[trace] 18 ?- mutter_von(julia,otto).
```

```
Call: (7) mutter_von(julia, otto) ? creep
```

```
Exit: (7) mutter_von(julia, otto) ? creep
```

```
true.
```

```
[trace] 19 ?- vater_von(otto, helga).
```

```
Call: (7) vater_von(otto, helga) ? creep
```

```
Exit: (7) vater_von(otto, helga) ? creep
```

```
true.
```

```
[trace] 20 ?- vater_von(X, julia).
```

```
Call: (7) vater_von(_G3242, julia) ? creep
```

```
Fail: (7) vater_von(_G3242, julia) ? creep
```

```
false.
```

```
[trace] 21 ?- vater_von(otto, X).
```

```
Call: (7) vater_von(otto, _G3237) ? creep
```

```
Exit: (7) vater_von(otto, hans) ? creep
```

```
X = hans ;
```

```
Redo: (7) vater_von(otto, _G3237) ? creep
```

```
Exit: (7) vater_von(otto, helga) ? creep
```

```
X = helga.
```

```
[trace] 22 ?- vater_von(X, Y);mutter_von(X, Y).
```

```
Call: (8) vater_von(_G3314, _G3315) ? creep
```

```
Exit: (8) vater_von(homer, lisa) ? creep
```

```
X = homer,
```

```
Y = lisa ;
```

```
Redo: (8) vater_von(_G3314, _G3315) ? creep
```

```
Exit: (8) vater_von(otto, hans) ? creep
```

```
X = otto,
```

```
Y = hans ;
```

```
Redo: (8) vater_von(_G3314, _G3315) ? creep
```

```
Exit: (8) vater_von(otto, helga) ? creep
```

```
X = otto,
```

```
Y = helga ;
  Redo: (8) vater_von(_G3314, _G3315) ? creep
  Exit: (8) vater_von(gerd, otto) ? creep
X = gerd,
Y = otto ;
  Redo: (8) vater_von(_G3314, _G3315) ? creep
  Exit: (8) vater_von(johannes, klaus) ? creep
X = johannes,
Y = klaus ;
  Redo: (8) vater_von(_G3314, _G3315) ? creep
  Exit: (8) vater_von(johannes, andrea) ? creep
X = johannes,
Y = andrea ;
  Redo: (8) vater_von(_G3314, _G3315) ? creep
  Exit: (8) vater_von(walter, barbara) ? creep
X = walter,
Y = barbara ;
  Redo: (8) vater_von(_G3314, _G3315) ? creep
  Exit: (8) vater_von(walter, magdalena) ? creep
X = walter,
Y = magdalena ;
  Redo: (8) vater_von(_G3314, _G3315) ? creep
  Exit: (8) vater_von(homer, bart) ? creep
X = homer,
Y = bart ;
  Call: (8) mutter_von(_G3314, _G3315) ? creep
  Exit: (8) mutter_von(marie, hans) ? creep
X = marie,
Y = hans ;
  Redo: (8) mutter_von(_G3314, _G3315) ? creep
  Exit: (8) mutter_von(marie, helga) ? creep
X = marie,
Y = helga ;
  Redo: (8) mutter_von(_G3314, _G3315) ? creep
  Exit: (8) mutter_von(julia, otto) ? creep
X = julia,
Y = otto ;
  Redo: (8) mutter_von(_G3314, _G3315) ? creep
  Exit: (8) mutter_von(barbara, klaus) ? creep
X = barbara,
Y = klaus ;
  Redo: (8) mutter_von(_G3314, _G3315) ? creep
  Exit: (8) mutter_von(barbara, andrea) ? creep
X = barbara,
```

```
Y = andrea ;
  Redo: (8) mutter_von(_G3314, _G3315) ? creep
  Exit: (8) mutter_von(charlotte, barbara) ? creep
X = charlotte,
Y = barbara ;
  Redo: (8) mutter_von(_G3314, _G3315) ? creep
  Exit: (8) mutter_von(charlotte, magdalena) ? creep
X = charlotte,
Y = magdalena ;
  Redo: (8) mutter_von(_G3314, _G3315) ? creep
  Exit: (8) mutter_von(marge, lisa) ? creep
X = marge,
Y = lisa ;
  Redo: (8) mutter_von(_G3314, _G3315) ? creep
  Exit: (8) mutter_von(marge, bart) ? creep
X = marge,
Y = bart.
```

```
[trace] 23 ?- \+ vater_von(hans,X).
  Call: (8) vater_von(hans, _G3255) ? creep
  Fail: (8) vater_von(hans, _G3255) ? creep
true.
```

```
[trace] 24 ?- \+ vater_von(johannes,X).
  Call: (8) vater_von(johannes, _G3279) ? creep
  Exit: (8) vater_von(johannes, klaus) ? creep
false.
```

```
[trace] 25 ?- \+(\+ vater_von(otto,X)).
  Call: (8) vater_von(otto, _G3279) ? creep
  Exit: (8) vater_von(otto, hans) ? creep
true.
```

```
[trace] 26 ?- nodebug.
true.
```