Nerijus Neverdauskas 709178
Luhao Qi 708878

Übungszettel 1

Aufgabe 1

a) [familie].
b) listing/0 : Anzeige des gesamten Inhalts der aktuellen Wissensbasis
listing/1 :

```
?- listing(mutter_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).

true.

?-
```

listing/2:

```
?- 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).
```

c) ?- assert(mutter_von(laura, jonas)).
Im Grunde dasselbe wie assertz( ).

```
?- [familie].
true.

?- assert(mutter_von(lauras, jonas)).
true.

?- listing(mutter_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).
mutter_von(lauras, jonas).

true.
```

Nerijus Neverdauskas 709178
Luhao Qi 708878

?- asserta(mutter_von(laura, jonas)).

Setzt behauptetes Material *an den Anfang* der Datenbank.

```
?- [familie].
true.

?- asserta(mutter_von(laura,jonas)).
true.

?- listing(mutter_von).
:- dynamic mutter_von/2.

mutter_von(laura, jonas).
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).

true.
```

?- asserrtz(mutter_von(laura, jonas)).

Stellt behauptetes Material *an das Ende* der Datenbank.

```
?- [familie].
true.

?- assertz(mutter_von(laura, jonas)).
true.

?- listing(mutter_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).
mutter_von(laura, jonas).
```

Aufgabe 2

a)

(i) Ist Charlotte die Mutter von Barbara?

```
?- mutter_von(charlotte,barbara).
true.
```

Ja.

(ii) Heißt der Vater von Andrea Walter?

```
?- vater_von(walter,andrea).
false.
```

Nein.

(iii) Wie heißt die Mutter von Hans?

```
?- mutter_von(X,hans).
X = marie.
```

Sie heißt Marie.

Nerijus Neverdauskas 709178
Luhao Qi 708878

(iv) Wie heißt die Mutter von Marie?

```
?- mutter_von(X,marie).
false.
```

Das ist uns nicht gegeben.

(v) Welche Kinder hat Johannes?

```
?- vater_von(johannes, Kinder).
Kinder = klaus ;
Kinder = andrea.
```

Johannes Kinder sind Klaus und Andrea.

(vi) Wer hat welche Kinder?

```
?- mutter_von(Mutter,MKind);vater_von(Vater,VKind).
Mutter = marie,
MKind = hans ;
Mutter = marie,
MKind = helga ;
Mutter = julia,
MKind = otto ;
Mutter = barbara,
MKind = klaus ;
Mutter = barbara,
MKind = andrea ;
Mutter = charlotte,
MKind = barbara ;
Mutter = charlotte,
MKind = magdalena ;
Vater = otto,
VKind = hans ;
Vater = otto,
VKind = helga ;
Vater = gerd,
VKind = otto ;
Vater = johannes,
VKind = klaus ;
Vater = johannes,
VKind = andrea ;
Vater = walter,
VKind = barbara ;
Vater = walter,
VKind = magdalena.
```

b)

```
?- mutter_von(julia,Kind),(mutter_von(Kind,Enkel);vater_von(Kind,Enkel)).
Kind = otto,
Enkel = hans ;
Kind = otto,
Enkel = helga.
```

Nerijus Neverdauskas 709178
Luhao Qi 708878

c)

```
[trace]  ?- mutter_von(X,hans).
   Call: (10) mutter_von(_12750, hans) ? creep
   Exit: (10) mutter_von(marie, hans) ? creep
X = marie.

[trace]  ?-  mutter_von(charlotte,barbara).
   Call: (10) mutter_von(charlotte, barbara) ? creep
   Exit: (10) mutter_von(charlotte, barbara) ? creep
true.

[trace]  ?- vater_von(walter,andrea).
   Call: (10) vater_von(walter, andrea) ? creep
   Fail: (10) vater_von(walter, andrea) ? creep
false.

[trace]  ?-  mutter_von(X,hans).
   Call: (10) mutter_von(_22830, hans) ? creep
   Exit: (10) mutter_von(marie, hans) ? creep
X = marie.

[trace]  ?- mutter_von(X,marie).
   Call: (10) mutter_von(_26284, marie) ? creep
   Fail: (10) mutter_von(_26284, marie) ? creep
false.

[trace]  ?- vater_von(johannes, Kind).
   Call: (10) vater_von(johannes, _28476) ? creep
   Exit: (10) vater_von(johannes, klaus) ? creep
Kind = klaus ;
   Redo: (10) vater_von(johannes, _28476) ? creep
   Exit: (10) vater_von(johannes, andrea) ? creep
Kind = andrea.
```

```
[trace]  ?- mutter_von(Mutter,MKind);vater_von(Vater,VKind).
   Call: (11) mutter_von(_25682, _25684) ? creep
   Exit: (11) mutter_von(marie, hans) ? creep
Mutter = marie,
MKind = hans ;
   Redo: (11) mutter_von(_25682, _25684) ? creep
   Exit: (11) mutter_von(marie, helga) ? creep
Mutter = marie,
MKind = helga ;
   Redo: (11) mutter_von(_25682, _25684) ? creep
   Exit: (11) mutter_von(julia, otto) ? creep
Mutter = julia,
MKind = otto ;
   Redo: (11) mutter_von(_25682, _25684) ? creep
   Exit: (11) mutter_von(barbara, klaus) ? creep
Mutter = barbara,
MKind = klaus ;
   Redo: (11) mutter_von(_25682, _25684) ? creep
   Exit: (11) mutter_von(barbara, andrea) ? creep
Mutter = barbara,
MKind = andrea ;
   Redo: (11) mutter_von(_25682, _25684) ? creep
   Exit: (11) mutter_von(charlotte, barbara) ? creep
Mutter = charlotte,
MKind = barbara ;
   Redo: (11) mutter_von(_25682, _25684) ? creep
   Exit: (11) mutter_von(charlotte, magdalena) ? creep
Mutter = charlotte,
MKind = magdalena ;
   Call: (11) vater_von(_25688, _25690) ? creep
   Exit: (11) vater_von(otto, hans) ? creep
Vater = otto,
VKind = hans ;
   Redo: (11) vater_von(_25688, _25690) ? creep
   Exit: (11) vater_von(otto, helga) ? creep
Vater = otto,
VKind = helga ;
   Redo: (11) vater_von(_25688, _25690) ? creep
   Exit: (11) vater_von(gerd, otto) ? creep
Vater = gerd,
VKind = otto ;
   Redo: (11) vater_von(_25688, _25690) ? creep
   Exit: (11) vater_von(johannes, klaus) ? creep
Vater = johannes,
VKind = klaus ;
   Redo: (11) vater_von(_25688, _25690) ? creep
   Exit: (11) vater_von(johannes, andrea) ? creep
Vater = johannes,
VKind = andrea ;
   Redo: (11) vater_von(_25688, _25690) ? creep
   Exit: (11) vater_von(walter, barbara) ? creep
Vater = walter,
VKind = barbara ;
   Redo: (11) vater_von(_25688, _25690) ? creep
   Exit: (11) vater_von(walter, magdalena) ? creep
Vater = walter,
VKind = magdalena.
```

Nerijus Neverdauskas 709178
Luhao Qi 708878

Aufgabe 3

a)

```
?- vater_von(P1,X),mutter_von(X,P2).
P1 = walter,
X = barbara,
P2 = klaus ;
P1 = walter,
X = barbara,
P2 = andrea ;
false.
```

In diesem Fall ist P1 der Großvater der Kinder(P2) seiner Tochter.

b)

```
?- mutter_von(X,P1),mutter_von(X,P2),P1\=P2.
X = marie,
P1 = hans,
P2 = helga ;
X = marie,
P1 = helga,
P2 = hans ;
X = barbara,
P1 = klaus,
P2 = andrea ;
X = barbara,
P1 = andrea,
P2 = klaus ;
X = charlotte,
P1 = barbara,
P2 = magdalena ;
X = charlotte,
P1 = magdalena,
P2 = barbara ;
false.
```

P1 und P2 sind Kinder von X, die selbst Geschwister sind.

c)

```
?- mutter_von(X,P1),mutter_von(Y,X),mutter_von(Y,P2),X\=P2.
X = barbara,
P1 = klaus,
Y = charlotte,
P2 = magdalena ;
X = barbara,
P1 = andrea,
Y = charlotte,
P2 = magdalena ;
false.
```

P2 ist Onkel oder Tante (in diesem Fall Tante - Magdalena) von P1

Nerijus Neverdauskas 709178
Luhao Qi 708878

d)

```
?- vater_von(X,P1),mutter_von(Y,X),mutter_von(Y,Z),mutter_von(Z,P2),X\=Z.
false.
```

P1 is a child of X, where X is son of Y and a brother of Z(female), who has a child P2.
X(brother) and Z(sister) are siblings.
P1 und P2 sind Cousinen.

e)

```
?- mutter_von(X,P1),mutter_von(Y,P2),vater_von(Z,P1), vater_von(Z,P2),P1\=P2,X\=Y.
false.
```

P1 und P2 sind Halbgeschwister mit gleichem Vater(Z).