

SEIII - Logikprogrammierung

Übungsblatt 02-03

Nico Hahn 6990715 Hieu Nguyen 6632126

Aufgabe 1

1. Ein Prädikat umfasst eine Folge von Wörtern mit klar definierten Leerstellen welches zu einer wahren oder falschen Aussage ausgewertet werden kann. Eine Klausel bezeichnet eine Boolesche Funktion, welche nur aus distinktiven Verknüpfungen von Literalen entsteht.
2. Im Hinblick auf die Logikprogrammierung ist die Suche eine Vorgang, welcher eine im System eingebundene Datenstruktur sequenziell durchläuft und angefragt Ergebnisse ausgibt. Eine Variable macht es möglich, nach dem sie mit dem default Wert instanziiert wurde, Ergebnisse in ihr abzuspeichern und mit diesen Werten weiter arbeiten zu können. Eine Instanziierung wie schon aus dem vorherigen Satz hervorgehend, ist eine Zuweisung eines Wertes an eine Variable.

Aufgabe 2

% 2.1 Welche Haeuser stehen in der Bahnhofsstrasse?

```
bahnHofHaeuser(ObjNr, HausNr) :-  
    obj(ObjNr,_,bahnhofsstr,HausNr,_).
```

```
% Ausgabe:  
% ObjNr = 2,  
% HausNr = 27,  
% ObjNr = 3,  
% HausNr = 29,  
% ObjNr = 4,  
% HausNr = 28,  
% ObjNr = 5,  
% HausNr = 30,  
% ObjNr = 6,  
% HausNr = 26,
```

% 2.2 Welche Haeuser wurden vor 1950 gebaut?

```
altBau(ObjNr,StrName,HausNr,BauJahr) :-  
    obj(ObjNr,_,StrName,HausNr,BauJahr), BauJahr < 1950.
```

```

% Ausgabe:
% ObjNr = 2,
% StrName = bahnhofsstr,
% HausNr = 27,
% BauJahr = 1943 ;
% ObjNr = 5,
% StrName = bahnhofsstr,
% HausNr = 30,
% BauJahr = 1901 ;
% false.

% 2.3 Wer besitzt Haeuser, die mehr als 300.000 Euro wert sind?

teuerHausBesitzer(Besitzer, Preis) :-
    bew(_,_,_,Besitzer,Preis,_), Preis > 300000.

% Ausgabe:
% Besitzer = mueller,
% Preis = 315000 ;
% Besitzer = piepenbrink,
% Preis = 1500000.

% 2.4 Welche Haeuser wurden mit Gewinn weiterverkauft

gewinnVerkauft(ObjNr, StrName, HausNr) :-
    obj(ObjNr,_,StrName,HausNr,_),
    bew(_,ObjNr,_,Kaeufer,KaufPreis,_),
    bew(_,ObjNr,Kaeufer,_,VerkaufPreis,_),
    VerkaufPreis > KaufPreis.

% Ausgabe:
% ObjNr = 3,
% StrName = bahnhofsstr,
% HausNr = 29 ;
% ObjNr = 3,
% StrName = bahnhofsstr,
% HausNr = 29 ;
% false.

% 2.5 Welche Haeuser haben schon mehrfach den Besitzer gewechselt

mehrGekauft(ObjNr, StrName, HausNr) :-
    bew(_,ObjNr,_,Besitzer1,_,_),
    bew(_,ObjNr,_,Besitzer2,_,_),Besitzer1 \= Besitzer2,
    obj(ObjNr,_, StrName, HausNr, _).

```

```

% Ausgabe:
% ObjNr = 3,
% StrName = bahnhofsstr ;
% ObjNr = 3,
% StrName = bahnhofsstr ;
% false.

% 2.6
% TODO

```

Aufgabe 3

```

% 3.1 Umrechnen zwischen Dateinamen und Dateischlüssel
% fNameToFId(?FId,?FName)
fNameToFId(FId,FName) :- file(FId,_,FName,_,_,_).

% 3.2 Umrechnen zwischen Verzeichnisnamen und Verzeichnisschlüssel
% dNameToDId(?DId,?DName)
dNameToDId(DId,DName) :- directory(DId,DName,_,_,_).

% 3.3 Name und Schlüssel des zugehörigen Verzeichnisses der Datei
% fileToDir(+FileName, -DirId, -DirName)
fileToDir(FileName, DirId, DirName) :- file(_,DirId,FileName,_,_,_),
                                         directory(DirId,DirName,_,_,_).

% 3.4 Name und Schlüssel des übergeordneten Verzeichnisses
% parentDir(+DirName, -ParentId, -ParentName)
parentDir(DirName, ParentId, ParentName) :-
    directory(_,DirName,ParentId,_,_),
    directory(ParentId,ParentName,_,_,_).

```

Aufgabe 4

```

% 4.1 Dateinamen
% listFile(+DirId, -FileList)
listFile(DirId, FileList) :-
    findall(FileName,file(_,DirId,FileName,_,_,_),FileList).

% 4.2 Namen aller Unterverzeichnisse
% listSubdir(+ParentDirId, -DirList)
listSubdir(ParentDirId, DirList) :-
    findall(SubDir,directory(_,SubDir,ParentDirId,_,_),DirList).

```

```
% 4.3 Anzahl aller Dateien in einem bestimmten Verzeichnis
% countFiles(+DirId, -Count)
countFiles(DirId, Count) :- listFile(DirId, FileList),
                             length(FileList,Count).
```