

Logische Programmierung

Labor 3 : Datenbanken und Rekursion in Prolog

Inhalt

- Rekursion
- Aufgabe

Rekursion - Kurzgesagt

Rekursion - eine Funktion, die sich selbst aufrufen kann, bis das Ziel erfüllt ist

Rekursion in Prolog - erscheint, wenn ein Prädikat ein Ziel enthält, das sich auf sich selbst bezieht

mindestens zwei Dinge - Ein erster Fakt, der wie eine Stopbedingung wirkt und eine Regel, die sich selbst vereinfacht ruft.

Eine rekursive Regel darf sich niemals selbst mit den gleichen Argumenten aufrufen! ACHTUNG: Unendliche Schleife!

Rekursion - Beispiel

```
parent(john,paul).  
parent(paul,tom).  
parent(tom,mary).
```

```
ancestor(X,Y):-parent(X,Y).
```

```
/* If Y is a parent of X, then Y is an ancestor of X */
```

```
ancestor(X,Y):-parent(X,Z), ancestor(Z,Y).
```

```
/* if Y is an ancestor of Z and Z is a parent of X, then Y is an ancestor of X */
```

Aufgabe 1

Wir können, dass die folgende Wissensbasis ein Labyrinth ist, vorstellen.

Die Fakten bestimmen, welche Punkte verbunden sind, d.h., von welchen Punkten man in einem Schritt zu welchen anderen Punkten gelangen kann.

Ausserdem vorstellen, dass alle Wege Einbahnstrassen sind(so dass man nur in eine Richtung gehen kann; oder nur von dem Punkt1 zu dem Punkt2, aber nicht von 2 zu 1).

connected(1,2).

connected(3,4).

connected(5,6).

connected(7,8).

Aufgabe 1

connected(9,10).
connected(12,13).
connected(13,14).
connected(15,16).
connected(17,18).
connected(19,20).
connected(4,1).
connected(6,3).
connected(4,7).
connected(6,11).

Aufgabe 1

connected(14,9).
connected(11,15).
connected(16,12).
connected(14,17).
connected(16,19).

Schreiben Sie ein Prädikat “path/2”, der Ihnen sagt, von welche Punkte in Labyrinth zu andere Punkte gehen können (miteinander verketteten diese Verbindungen, die in unserer Wissensbasis sind).

Aufgabe 1

Können Sie von Punkt 5 zu Punkt 10 gehen?

Welche andere Punkte können Sie ankommen, wenn Sie mit Punkt 1 beginnen?

Und welche Punkte können Sie ankommen, wenn Sie mit Punkt 13 beginnen?

Aufgabe 2

Wir bekommen die folgende Reiseinformationen Wissensbasis:

byCar(auckland,hamilton).

byCar(hamilton,raglan).

byCar(valmont,saarbruecken).

byCar(valmont,metz).

byTrain(metz,frankfurt).

byTrain(saarbruecken,frankfurt).

byTrain(metz,paris).

byTrain(saarbruecken,paris).

Aufgabe 2

byPlane(frankfurt,bangkok).

byPlane(frankfurt,singapore).

byPlane(paris,losAngeles).

byPlane(bangkok,auckland).

byPlane(singapore,auckland).

byPlane(losAngeles,auckland).

Aufgabe 2

Schreiben Sie ein Prädikat “travel/2” dass, ob es möglich von einem Punkt bis einen anderen Punkt bei Auto, Zug, Flugzeug oder Kombinationen von diese vorherige zu reisen, bestimmt.

Z.B. Ihr Programm sollte das Antwort “yes” für die Anfrage “travel(valmont,raglan)” zurückgeben.

Aufgabe 3

Betrachten Sie die folgende Wissensbasis:

et(albert,kevin).

et(lena,albert).

et(marie,lena).

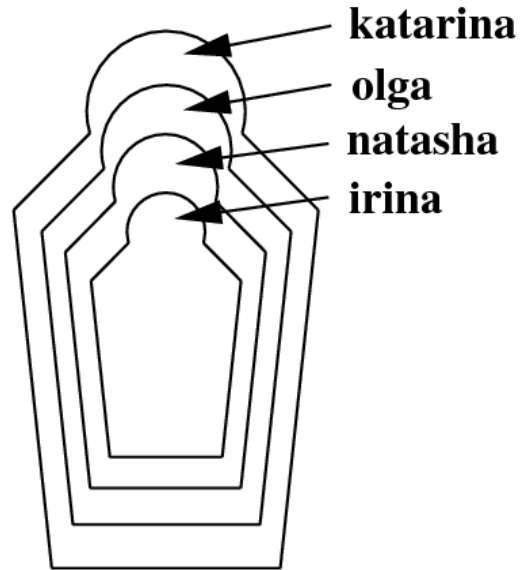
Betrachten Sie die folgende Definitionsvariante für das Prädikat 'vorfahr/2'. Welche Probleme ergeben sich für diese Variante ?

vorfahr(X,Y) :- et(X,Y).

vorfahr(X,Y) :- vorfahr(X,Z), vorfahr(Z,Y).

Aufgabe 4

Wissen Sie die russische Matroschka Puppen? Checken das folgende Bild:



Aufgabe 4

Schreiben Sie eine Wissensbasis mit Hilfe von den Prädikat 'directlyin/2'. Dies Prädikat sagt ob eine Puppe in eine andere Puppe ist. Definieren Sie ein rekursiv Prädikat 'in/2'. Es sagt ob eine Puppe in eine andere Puppe (direkt oder indirekt) ist.

z.B. `in(katarina,natasha) => false ; in(olga,katarina) => true.`

Aufgabe 5-6

5. Schreiben Sie ein Prolog Programm, das bis zehn absteigend zählt.
6. Schreiben Sie ein Prolog Programm, das zwei Zahlen X und Y vergleicht. Dies Programm zeigt die Nachricht "X ist grosser oder egal" wenn X grosser oder egal mit Y ist, oder zeigt die Nachricht "X ist kleiner" wenn $X < Y$.