Logikprogrammieren Hausaufgaben

Blatt 3

Morten Seemann 6945442, Tore Wiedenmann 6488837

Aufgabe 1: Definieren Sie folgende Prädikate auf der Basis der Peano-Arithmetik. Testen Sie sie mit geeigneten Beispielen.

1.) Ein Prädikat, das eine Peano-Zahl in eine Integer-Zahl umwandelt.

```
peano2int(0,0).
peano2int(s(X),INT) :- peano2int(X,I1),INT is I1 +1.

%TEST
?- peano2int(s(s(s(0))),X).
X = 3.
```

2.) Ein Prädikat, das den unmittelbaren Nachfolger einer Peano-Zahl ermittelt

```
suc(0,s(0)).
suc(s(X),SUC) :- add(s(X),s(0),SUC).

%TEST
?- suc(s(0),X).
X = s(s(0)).
```

3.) Ein Prädikat, das den unmittelbaren Vorgänger einer Peano-Zahl ermittelt

```
prev(s(0),0).
prev(s(X),PREV):- add(PREV,s(0),s(X)).

%TESTS
?- prev(0,X).
false.

?- prev(s(s(0)),X).
X = s(0) .
```

4.) ein Prädikat für die Subtraktion zweier Peano-Zahlen.

```
%Wenn X - Y = R  dann ist R+Y=X
subtract(s(X),s(Y),s(R)):- add(s(R),s(Y),s(X)).

%TEST
?- subtract(s(s(s(s(0)))),s(s(0)),X).
X = s(s(0)) .
```

5.) ein rekursives Prädikat "min(?Peano1,?Peano2,?PeanoMin)", das für zwei Peano-Zahlen Peano1 und Peano2 deren Minimum als PeanoMin ermittelt.

```
minimum(0,_,0).
minimum(s(X),s(Y),s(R)):-lt(s(X),s(Y)),s(R)=s(X).

min(X,Y,X):- minimum(X,Y,X).
min(X,Y,Y):- minimum(Y,X,Y).
```

```

%TESTS
?- min(0,s(s(0)),X).
X = 0 .

?- min(s(s(s(0))),s(s(0)),X).
X = s(s(0)).

?- min(s(s(0)),0,X).
X = 0.
```

6. Ein Prädikat für die Multiplikation zweier Peano-Zahlen

```
% 0*x=0
mul(0,_,0).
mul(_,0,0).
% 1*x=x
mul(s(0),X,X).
mul(X,s(0),X).
% n*x = x + (n-1)*x
mul(s(X),s(Y),s(P)):-add(s(P1),s(Y),s(P)),mul(X,s(Y),s(P1)).

%TEST

?- mul(s(s(s(0))),s(s(0)),X).
X = s(s(s(s(s(s(0)))))) .
```

Aufgabe 2: 1.)Modifizieren Sie das Prädikat ist erreichbar/3 von Aufgabenblatt 4 so, dass auch die Länge einer Strecke zwischen zwei Punkten ermittelt werden kann.

```
ist_erreichbar(S,Z,M) :- strecke(_,S,Z,_,M).
ist_erreichbar(S,Z,M) :- strecke(_,S,X,_,M1),
                              ist_erreichbar(X,Z,M2),
                               M is M1+M2.
```

Diese Lösung ist nicht endrekursiv da das Ergebniss erst ganz zum Schluss berechnet wird und nicht bereits Zwischenergebnisse dem weiteren Aufrufen Übergeben werden.

2.)Verwenden Sie das Prädikat aus der vorangegangenen Aufgabe, um die Länge des längsten, zusammenhängend befahrbaren Weges zwischen zwei Punkten zu ermitteln. Überlegen Sie sich zwei alternative algorithmische Lösungsansätze und diskutieren Sie die beiden Varianten im Hinblick auf ihr Berechnungsverhalten.

```
%Ansatz 1: Alle Strecken X->Y in einer Liste sammeln,sortieren und das
   groesste waehlen.

%Ansatz 2: Die Laengste Strecke X->Y ist genau die fuer die es keine
   laengere Strecke gibt.
laengste_strecke(X,Y,M):- ist_erreichbar(X,Y,M),
                              not(( ist_erreichbar(_,_,M2),
                                    M2>M)).
```

In Ansatz 1 Wird jede Strecke einmal in die Liste eingefügt und dann wird die Gesamte Liste einmalig Sortiert. In Ansatz 2 wird jede Strecke beim erstmaligen auswählen mit allen anderen Strecken verglichen um entscheiden zu können ob diese Ausscheiden muss. Wenn der Vergleich bei keiner Strecke fehlschlägt terminiert Ansatz 2. Ansatz 2 zwei terminiert dabei schneller als Ansatz 1 da das aggregieren der Strecken nicht notwendig ist und Ansatz 2 quasi bereits der Sortierschritt aus Ansatz 1 ist.