Logikprogrammieren Hausaufgaben

Blatt 8

Morten Seemann 6945442, Tore Wiedenmann 6488837

Aufgabe 1: 1.) Modifizieren Sie das Prädikat ist erreichbar/3 von Aufgabenblatt 4 so, dass auch die Wegstrecke zwischen Startpunkt und Ziel als Liste von Knotenpunkten im Skigebiet berechnet werden kann.

```
ist_erreichbar(S,Z,[S,Z]):-strecke(_,S,Z,_,_).

ist_erreichbar(S,Z,[S|L1]):-strecke(_,S,X,_,_),
                            ist_erreichbar(X,Z,L1).
```

2.) Erweitern Sie das Prädikat aus Teilaufgabe 1 so, dass auch die Benutzung der Seilbahnen berücksichtigt werden kann. Begrenzen Sie dabei die Suchtiefe durch eine maximale Anzahl von Fahrten mit den Seilbahnen.

```
ist_erreichbar(S,Z,[S,Z]):-strecke(_,S,Z,_,_).
ist_erreichbar(S,Z,[X,Z]):- lift(X,S,Z,_).

ist_erreichbar(S,Z,[S|L1]):-strecke(_,S,X,_,_),
                            ist_erreichbar(X,Z,L1).


% Wenn man 1 mal den Lift benutzen darf, so sucht man eine normale Strecke
ist_erreichbar(S,Z,L,1):- ist_erreichbar(S,Z,L).

% Man darf N mal lift fahren, man 1 mal und N-1 mal lift fahren darf.
ist_erreichbar(S,Z,Liste,N):- N>1,
                              N1 is N -1,
                              ist_erreichbar(S,Z,L,1), %Ein mal
                              ist_erreichbar(S,Z,L1,N1), %N-1 mal
                              append(L,L1,Liste).
```

3.) 3. Modifizieren Sie das Prädikat aus Teilaufgabe 2 so, dass eine Seilbahn nicht mehrfach benutzt werden kann.

```
% Gibt einen Lift an den eine Liste von Knoten hat.
hat_lift(L,X):- lift(X,_,_,_),member(X,L).

% Die NAND Klausel mit hat_lift ist nur dann war wenn nicht der gleiche
   Lift in beiden Listen Steht.
ist_erreichbar(S,Z,Liste,N):- N>1,
                              N1 is N -1,
                              ist_erreichbar(S,Z,L,1),
                              ist_erreichbar(S,Z,L1,N1),
                              not((hat_lift(L,X),hat_lift(L1,X))),
                              append(L,L1,Liste).

%Testbar z.B. mit ist_erreichbar(kn1,bgpanorama,X,2).
%(Achtung, sehr viele Ergebnisse)
```

4.) Implementieren Sie eine Lösung, die nur solche Routen durch das Skigebiet berechnet, die man innerhalb einer vorgegebenen maximalen Zeitspanne durchfahren kann.

```
%Die Modifizierte Version von ist_erreichbar gibt zusatzlich noch die Zeit
    in Sekunden an
%Die eine solche Strecke braucht.

ist_erreichbar(S,Z,[S,Z],T):-strecke(_,S,Z,_,T1),T is T1/4.
%oder direkter Lift in T1 minuten
ist_erreichbar(S,Z,[X,Z],T):- lift(X,S,Z,T1),T is T1*60.

%Oder durch eine Strecke zu X s.d. man von X zum Ziel kommt
ist_erreichbar(S,Z,[S|L1],T):-strecke(_,S,X,_,T0),
                              T1 is T0/4,
                              ist_erreichbar(X,Z,L1,T2),
                              T is T1+T2.


%Wenn man 1 mal den Lift benutzen darf, so sucht man
%eine normale Strecke
ist_erreichbar(S,Z,L,1,T,MaxT):- ist_erreichbar(S,Z,L,T),T<MaxT.
% Man darf N mal lift fahren wenn,
% man 1 mal und N-1 mal lift fahren darf.
ist_erreichbar(S,Z,Liste,N,T,MaxT):- N>1,
                                  N1 is N -1,
                                  ist_erreichbar(S,Z,L,1,T1,MaxT),
                                  ist_erreichbar(S,Z,L1,N1,T2,MaxT),
                                  T is T1+T2,
                                  T<MaxT
                                 % not((hat_lift(L,X),hat_lift(L1,X))),
                                  append(L,L1,Liste).
%Tests
%Die Strecken + Dauer in Sekunden.
L = [kn1,kn2,kn3,kn4,bggondel,rootmoos,bgrootmoos],
T = 1395 ;
L = [kn1,kn6,kn3,kn4,bggondel,rootmoos,bgrootmoos],
T = 1270 ;
L = [kn1,kn6,kn7,bggondel,rootmoos,bgrootmoos],
T = 1295 ;
L = [kn1,kn6,kn7,bggondel,rootmoos,bgrootmoos],
T = 1270 ;
L = [kn1,kn7,bggondel,rootmoos,bgrootmoos],
T = 1120 ;
L = [kn1,kn7,bggondel,rootmoos,bgrootmoos],
T = 1095 ;

?- ist_erreichbar(kn1,bgrootmoos,L,1,T,1200).
L = [kn1,kn7,bggondel,rootmoos,bgrootmoos],
T = 1120 ;
L = [kn1,kn7,bggondel,rootmoos,bgrootmoos],
T = 1095 ;
```

Bonus 1.)

```
% Hans befindet sich um 13 Uhr...

% Hans muss mit dem Lift nach bggondel starten.
% Dieser kostet 900 sekunden.
% Danach kann er eine Strecke Waehlen die maximal 2700 sekunden braucht,
% und beliebeig( hier bis zu 6 mal da N instanziiert sein muss) einen Lift benutzt.
hans(L,T):- member(N,[1,2,3,4,5,6]),
            ist_erreichbar(bggondel,tlgondel,L,N,T,2700).

% Hans ist unzufrieden...

% Sammelt alle (Restzeit,Liste) paare und sortiert nach geringster Restzeit.
hans_sortiert(L):- findall((T,X),(hans(X,T1),T is 2700-T1),L1),sort(L1,L).
```

Aufgabe 2: 1.) Implementieren Sie ein Prädikat, das für ein gegebenes Münzsystem (z.B. die Euro-Munzen) alle möglichen Zerlegungen des Wechselgeldes als Liste von Munzwerten berechnet.

```
muenze(X):-member(X,[1,2,5,10,20,50,100,200]).

wechselgeld(X,[X]):-muenze(X).

wechselgeld(Sum,[H|T]):- Sum>1,
                         muenze(H),
                         S2 is Sum-H,
                         wechselgeld(S2,T).
```

2.)Stellen Sie in Ihrer Implementation sicher, dass Ihr Prädikat zwar alle möglichen Zerlegungen des Betrags in Münzen ermittelt, aber keine Mehrfachresultate, d.h. auch keine Permutationen der Ergebnisliste ausgibt. Begründen Sie Ihre Entscheidung für den von Ihnen verwendeten Lösungsansatz.

```

%msort ist sort ohne das entfernen von duplikaten.
sortiertes_wechselgeld(S,L):- wechselgeld(S,A),msort(A,L).

% Wenn alle moeglichen sortierten Listen gefunden werden muessen gleiche
% Listen jedoch wieder entfernt werden.
moegliches_wechselgeld(S,L):- findall(K,sortiertes_wechselgeld(S,K),L1),sort(L1,L).

% Ein Wrapper zu konstruieren welcher alle Listen bereinigt schien die intuitive Loesung zu sein.
% Das haengt natuerlich vom Typ des gesuchten Ergebnisses ab, wenn alle Listen nur auf erneute Abfrage
% ausgegeben werden sollen verfehlt der wrapper das Ziel.
% Eine Liste von Listen ist oft jedoch nuetzlicher zur weiterverarbeitung.
```

3.)Modifizieren Sie Ihre Prädikatsdefinition so, dass fur das System der Euromuenzen immer nur die optimale Zerlegung fuer das Wechselgeld ermittelt wird, d.h. diejenige, die mit der kleinsten Zahl von Munzen auskommt.

```
% Es wird die Groesste moegliche muenze ausgewaehlt

max_kandidat(S,K):- muenze(K),S1 is S-K,not(S1<0),
                    not((muenze(K1),K1>K,S2 is S-K1,not(S2<0),S2<S1)).

% Das optimale wechselgeld besteht nur aus den groesstmoeglichen muenzen.
wechselgeld2(X,[X]):-max_kandidat(X,X).
wechselgeld2(Sum,[H|T]):- Sum>1,
                          max_kandidat(Sum,H),
                          S2 is Sum-H,
                          wechselgeld2(S2,T).
```

4.)Erweitern Sie Ihre Prädikatsdefinition so, dass sie in einem Automaten eingesetzt werden kann und bei der Herausgabe des Wechselgeldes berucksichtigt, ob die jeweils benötigten Münzen im Geldspeicher noch vorhanden sind. Ihr Prädikat sollte auch berechnen, wieviele Münzen der verschiedenen Sorten der Münzspeicher nach Herausgabe des Geldes noch enthält.

```
% Wenn M eine Muenzgroesse und Menge eine Menge solcher dann ist M
   enthalten falls es nicht
% 0 Muenzen von dieser Groesse gibt.
allgemeine_muenze(M,Menge):- member((M,X),Menge),X>0.

% Ein Maximaler Kandidat ist dann einer den es in der Menge auch
   tatsaechlich gibt.
max_kandidat(S,K,M):- allgemeine_muenze(K,M),S1 is S-K,not(S1<0),
                    not((allgemeine_muenze(K1,M),K1>K,S2 is S-K1,not(S2<0)
   ,S2<S1)).

%Und Wechselgeld soll nur aus der angegeben Menge rausgegeben werden.
wechselgeld2(X,[X],M):-max_kandidat(X,X,M).
wechselgeld2(Sum,[H|T],M):- Sum>1,
                          max_kandidat(Sum,H,M),
                          S2 is Sum-H,
                          wechselgeld2(S2,T,M).


% automat(+Wechselgeld,+Muenzvorat,?NeuerMuenzvorat).
automat(L,L,[]).
automat([],L,L).
automat([H|T],V,Vneu):- member((H,X),V),
                        X>0,
                        subtract(V,[(H,X)],V1),
                        X1 is X-1,
                        append(V1,[(H,X1)],V2),
                        automat(T,V2,Vneu).

% Unser Automat kann den Betrag S wechseln mit L als Wechselgeldliste und
   V als Vorat und hat
% nach dem Wechsel noch Vneu als Vorat.
automat_wechselgeld(S,L,V,Vneu):- wechselgeld2(S,L,M),automat(L,V,Vneu).
```

```
35
36 %TEST, 15 cent herausgeben aus einem Automaten mit 100 1 cent stuecken,
      einem 1 Euro stueck und Null 5 cent
37 ?- automat_wechselgeld(15,L,[(1,100),(20,1),(5,0)],Neu).
38 L = [1, 1, 1, 1, 1, 1, 1, 1, 1|...],
39 Neu = [(20, 1),  (5, 0),  (1, 85)].
40
41 %liefert das gewuenschte ergebniss.
```

Bonus.) Erweitern Sie Ihr Prädikat aus der vorangegengenen Teilaufgabe zu einem kompletten Warenautomaten,...

```
1  % bonus(+Preis,+Gegeben,?Wechsel,+Vorat,?VoratNeu).
2
3  bonus(Preis,Gegeben,Wechsel,Vorat,VoratNeu)
4              :- Rueckgeld is Gegeben-Preis,
5                 automat_wechselgeld(Rueckgeld,Wechsel,Vorat,VoratNeu).
6
7  %Test
8  ?- bonus(32,40,L,[(1,100),(20,1),(5,0)],Vneu).
9  L = [1, 1, 1, 1, 1, 1, 1, 1],
10 Vneu = [(20, 1),  (5, 0),  (1, 92)] ;
11
12 ?- bonus(32,40,L,[(1,100),(20,1),(5,1)],Vneu).
13 L = [5, 1, 1, 1],
14 Vneu = [(20, 1),  (5, 0),  (1, 97)] ;
```