Aufgabenblatt 6: Einführung in das Symbolische Programmieren CIS, WS 2010
==========

Abgabe: Am Montag, 29.11. 8.00 Uhr durch die Abgabemaske auf der Kursseite

Aufgabe 13

13.1) Gegeben seinen zwei Listen, L1 und L2. Schreiben Sie ein Prädikat durchschnitt/3, das eine Liste erzeugt, in der nur Elemente vorkommen, die sowohl in L1 als auch in L2 vorkommen.
L1=[1,2,3,4],L2=[3,4,5,6],L3=[3,4]
13.2) Implementieren Sie nun das Prädikat differenz/3 indem nur Elemente vorkommen die exklusiv in L1 vorkommen.
L1=[1,2,3,4],L2=[3,4,5,6],L3=[1,2]
13.3) Implementieren Sie ein Prädikat vereinigungsmenge/3, das die Vereinigungsmenge von zwei Listen erstellt. Sie dürfen annehmen, dass die Eingabelisten Mengen sind.
13.4) Implementieren Sie nun das Prädikat durchschnittsmenge/3, das die Durchschnittsmenge von zwei Listen erstellt. Wiederum dürfen Sie annehmen, dass die Eingabelisten Mengen sind. Überlegen Sie ob es einen Unterschied zu Aufgabe 13.1 gibt.

14.1) Gegeben sei ein kreisfreier Graph mit den Knoten 1 -7. Die Kanten des Graphen sollen
kante(1,2).
kante(1,3)
kante(2,4)
kante(2,5)
kante(3,4)
kante(4,7)
kante(5,6)
kante(6,7)

Schreiben Sie ein Prädikat pfad, das alle Pfade im Graphen berechnet und als Listen ausgibt: Beispielausgabe X = [1,2,4,7]. Mit ausgelöstem Backtracking sollen alle Pfade im Graphen ausgegeben werden können. Es hilft, wenn Sie den Graphen aufzeichnen. Zusätzlich zum rekursiven Pfadprädikat müssen Sie eine Liste mit den Knoten des Pfads erzeugen.

14.2) Benutzen Sie eine Schleifenkonstruktion wie in der Vorlesung vorgestellt um alle Wege auszugeben. Hinweise in drucke(X) hatten wir schon das write Prädikat benutzt.

14.3) Fügen Sie die folgenden Kanten ein:
kante(6,1).
kante(7,2).
kante(7,3).
Der Graph ist nun zyklisch. Schreiben Sie ein Programm, das alle zyklenfreien Pfade des Graphen in der Form von Listen berechnet. Hinweis: Die Listen dürfen keine doppelten Elemente enthalten.

15.1) Betrachten Sie folgendes Prolog Programm:

p(1).
p(2):-!.
p(3).

Geben Sie die Antworten folgender Anfragen an:

(a) ?- p(X).
(b) ?- p(X),p(Y).
(c) ?- p(X),!,p(Y).

15.2)
Definieren Sie eine Relation, die eine EingabeZahl in drei mögliche Klassen einteilt: Positive (X>0), Null (X /==0) und Negative (X<0). Benutzen Sie den cut, um das Programm möglichst effizient zu machen.