SE3	# Aufgabenblatt 6 Softwareentwicklung 3: Logikprogrammierung - WS 2019/2020 - W. Menzel Rekursive Berechnungen/rekursive Strukturen

Gesamtpunktzahl: 30 Abgabe der Lösungen bis zum 2.12.2019

Hinweis: Bitte beachten Sie bei der Bearbeitung der Aufgaben, dass Sie

- Prädikatsdefinitionen immer übersichtlich strukturieren und ausführlich kommentieren,
- in jedem Fall ein Prädikatsschema mit Zusicherungen für die zulässigen Datentypen und den möglichen Instanziierungsvarianten für die einzelnen Argumentpositionen angeben und
- die von Ihnen durchgeführten Tests mit ihren jeweiligen Resultaten dokumentieren und ggf. diskutieren.

Aufgabe 1: Rekursive Berechnungsvorschriften 10 Punkte
maximale Bearbeitungszeit: 40 Minuten

Die Zahl π kann durch die LEIBNITZ'sche Reihenzerlegung berechnet werden:

$$\pi = 4(1 - \frac{1}{3} + \frac{1}{5} - \ldots) = 4 \sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{2n-1}$$

1. Definieren Sie ein Prädikat, das π mit einer vorgegebenen Anzahl von Rekursionsschritten approximiert, z.B.

   ```
   ?- pi(10,Resultat).
      Resultat = 3.23232 ;
      No
   ```

   Implementieren Sie Ihr Prädikat in zwei Varianten, wobei die Berechnung einmal beim rekursiven Abstieg und zum anderen beim rekursiven Aufstieg erfolgen soll. In welchem Fall liegt Endrekursion vor?

2. Vergleichen Sie Ihre Definitionen im Hinblick auf die Verständlichkeit und das Berechnungsverhalten.

3. Implementieren Sie eine alternative Variante zur Berechnung von π mit Hilfe der WALLIS'schen Formel:

$$\frac{\pi}{2} = \prod_{i=1}^{\infty} \frac{2i}{2i-1} \cdot \frac{2i}{2i+1}$$

Vergleichen Sie Genauigkeit und Rechenzeitbedarf mit den Implementationen aus Teilaufgabe 1.

Aufgabe 2: Stromorientierte Verarbeitung 7 Punkte
maximale Bearbeitungszeit: 40 Minuten

1. Implementieren Sie auf der Basis des Rekursionsschemas für die stromorientierte Verarbeitung inkrementelle Varianten der in Aufgabe 1 entwickelten Prädikate für die Approximation von π nach Leibnitz und Wallis, die die aktuellen Approximationswerte als Folge von Lösungsalternativen über das Backtracking ermitteln, z.B.:

```
?- pi_incr(3,Pi).
   Pi = 4 ;
   Pi = 2.66667 ;
   Pi = 3.46667 ;
   Pi = 2.89524 ;
No
```

2. Analysieren Sie das Konvergenzverhalten der beiden Berechnungsvorschriften für π. Diskutieren Sie die Unterschiede.

   Bei Bedarf können Sie sich mit Hilfe des in der Datei `display.pl` definierten Prädikats den zeitlichen Verlauf der Reihenzerlegung durch den Aufruf des Ziels

   ```
   ?- findall(X,pi_incr(150,X),L),display('pi',L).
   ```

   darstellen lassen. Dies funktioniert leider nicht unter MacOS.

Aufgabe 3: verzweigende rekursive Strukturen 13 Punkte
maximale Bearbeitungszeit: 60 Minuten

Wir betrachten rekursiv eingebettete Strukturen (Binärbäume) des Typs `s(a,b)`, `s(s(a,b),c)`, `s(a,s(b,c))`, `...` die nur aus zweistelligen Strukturen mit dem Funktor `s` bzw. aus Namen zusammengesetzt sind.

1. Definieren Sie einen Typtest für derartige Strukturen.

   Hinweis: Orientieren Sie sich am Typtest für die Peano-Zahlen. Verwenden Sie für den Test auf das Vorliegen eines Namens das Prädikat `atom/1`.

2. Definieren Sie ein nicht-endrekursives und ein endrekursives Prädikat, das für einen Binärbaum die lokalen Einbettungstiefen auf allen Pfaden vom Wurzelknoten zu den einzelnen Blattknoten als alternative Variablenbindungen ermittelt.

   Hinweis: Orientieren Sie sich an den Prädikaten zur Umwandlung von PEANO-Zahlen. Verwenden Sie separate Klauseln für den linken und den rechten Zweig jedes Teilbaumes.

3. Verwenden Sie das Prädikat aus Teilaufgabe 2, um die maximale Einbettungstiefe eines Binärbaums zu ermitteln.

4. Definieren Sie ein alternatives, *rekursives* Prädikat, das für einen Binärbaum die maximale Einbettungstiefe ermittelt.

   Hinweis: Fassen Sie die beiden rekursiven Klauseln aus einer der Lösungen für Aufgabenteil 2 zu einer einzigen Klausel zusammen und vergleichen Sie in dieser Klausel die lokalen Tiefenangaben aus dem linken und rechten Zweig des Baumes.

   Überlegen Sie sich, welche der beiden Definitionsvarianten (endrekursiv bzw. nicht-endrekursiv) auf eine einfachere Lösung führt?

5. Definieren Sie ein *rekursives* Testprädikat, das überprüft, ob ein Binärbaum balanciert ist, d.h. die lokalen Einbettungstiefen dürfen sich maximal um den Wert eins unterscheiden.

   Hinweis: Erweitern Sie die Lösung zu Aufgabenteil 4 so, dass das Prädikat zusätzlich auch die minimale Einbettungstiefe für einen (Teil-)Baum ermittelt und stellen Sie sicher, dass der Unterschied zwischen Minimum und Maximum unterhalb der geforderten Schwelle bleibt.