SE3 Aufgabenblatt 6

Softwareentwicklung 3: Logikprogrammierung - WS 2014/2015 - W. Menzel

Rekursive Berechnungen

Gesamtpunktzahl: 30 Abgabe der Lösungen bis zum 1.12.2014

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: Numerische Rekursion (2) 8 Punkte
maximale Bearbeitungszeit: 40 Minuten

Der goldene Schnitt ist das Längenverhältnis zweier Strecken a und b, für die gilt

$$\Phi = \frac{a}{b} = \frac{a+b}{a}.$$

Eine geometrische Aufteilung in diesem Verhältnis wird von vielen Menschen als besonders harmonisch empfunden und spielt insbesondere in der klassischen Architektur und Malerei ein wichtige Rolle.

Der goldene Schnitt kann durch folgende rekursive Berechnungsvorschrift approximiert werden:

$$\Phi_{n+1} = \frac{1}{\Phi_n} + 1$$

1. Definieren Sie ein Prädikat, das den goldenen Schnitt mit einer vorgegebenen Anzahl von Rekursionsschritten berechnet, z.B.

   ```
   ?- goldener_schnitt(10,Resultat).
      Resultat = 1.6180555555555556 ;
      false.
   ```

   Ermitteln Sie einen geeigneten Startwert für die rekursive Approximation, der eine möglichst schnelle Konvergenz erlaubt.

   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.

   Hinweise: Zur Überprüfung der Ergebnisse können Sie die Darstellungsgenauigkeit für Gleitkommazahlen erhöhen, indem Sie

   ```
   ?- set_prolog_flag(float_format,'%.30g').
   ```

   am Prompt des Prolog-Systems eingeben. Den Rechenzeitbedarf eines Programmlaufs könnern Sie ermitteln, wenn Sie den Aufruf in das einstellige Prädikat `time` einbetten.

3. Ermitteln Sie, wie viele Approximationsschritte erforderlich sind, um den goldenen Schnitt mit einer Genauigkeit von 10^{-10} zu berechnen.

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

1. Implementieren Sie eine rekursive Prozedur `nat_zahl/1`, die über das Backtracking alternative Wertebindungen für ihr Argument erzeugt, so dass eine Folge von natürlichen Zahlen entsteht:

   ```
   ?- nat_zahl(Resultat).
   Resultat = 0 ;
   Resultat = 1 ;
   Resultat = 2
   ```

   Hinweis: Definieren Sie dazu ein Hilfsprädikat, das in einem Akkumulator auf einer zusätzlichen Argumentstelle die maximale, bisher erzeugte natürliche Zahl verwaltet.

2. Modifizieren Sie Ihr Prädikat so, dass eine maximale Obergrenze für die zu erzeugenden natürlichen Zahlen angegeben werden kann.

3. Implementieren Sie analog zur vorangegangenen Teilaufgabe eine inkrementelle Variante des Prädikats für den goldenen Schnitt, die ebenfalls die aktuellen Approximationswerte als Folge von Lösungsalternativen über das Backtracking ermittelt, z.B.:

   ```
   ?- goldener_schnitt_incr(3,R).
      R = 2 ;
      R = 1.5 ;
      R = 1.6666666666666665 ;
      R = 1.6 ;
      false.
   ```

4. Veranschaulichen Sie sich das Konvergenzverhalten der Berechnungsvorschrift für den goldenen Schnitt grafisch. Laden Sie dazu die Datei `display.pl`. Den zeitlichen Verlauf der Reihenzerlegung können Sie sich dann durch den Aufruf des Ziels

   ```
   ?- findall(X,goldener_schnitt_incr(15,X),L),
      display('Goldener Schnitt',L).
   ```

   darstellen lassen.

   Wieviele Approximationsschritte sind erforderlich, damit bei der gegebenen Auflösung der Approximationsfehler unter die Darstellungsgenauigkeit sinkt?

Aufgabe 3: verzweigende rekursive Strukturen 15 Punkte
maximale Bearbeitungszeit: 70 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 Spitzenknoten 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 des Baumes.

3. Definieren Sie ein *nichtrekursives* Prädikat, das für einen Binärbaum die maximale Einbettungstiefe ermittelt. Sammeln Sie dazu alle lokalen Einbettungstiefen in einer Liste auf und wählen sie daraus das maximale Element. Informieren Sie sich im Handbuch über Prädikate zur Berechnung des Maximums für eine gegebene Liste von Zahlen.

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.