

Ausgabe ..... Donnerstag, den 20.11.2003  
 Abgabe der Lösungen ..... 28.11.2003 bzw. nach Vereinbarung mit dem ÜG-Leiter

Homepage von Leonie Dreschler-Fischer ..... <http://kogs-www.informatik.uni-hamburg.de/~dreschle/home.html>  
 Homepage von Wolfgang Menzel ..... <http://nats-www.informatik.uni-hamburg.de/~wolfgang/>  
 P1-Homepage ..... <http://nats-www.informatik.uni-hamburg.de/p1>

Bitte denken Sie daran, auf die von Ihnen eingereichten Lösungsvorschläge *Name*, *Name des Übungsleiters* und *Wochentag und Uhrzeit der Übungsgruppe* anzugeben, damit wir ihre Ausarbeitungen eindeutig zuordnen können.

Die nachfolgenden Aufgaben üben den Entwurf, die Implementation und die Bewertung von rekursiven Berechnungsvorschriften.

Der Zeitaufwand für die Bearbeitung dieses Aufgabenblatts sollte 150 Minuten nicht übersteigen.

**1. Berechnung einer geometrischen Reihe** (8 Punkte, Zeitbedarf maximal 30 Minuten)

- (a) Definieren Sie auf der Basis der arithmetischen Auswertungsumgebung ein vierstelliges Prädikat für die Berechnung des Zinseszins mit den folgenden Argumenten
- i. Anlagebetrag
  - ii. Zinsrate
  - iii. Anlagezeitraum (in Jahren)
  - iv. Endbetrag
- (b) Schätzen Sie durch systematisches Probieren die Zinsrate, die benötigt wird, um in 10 Jahren eine Kapitalverdopplung zu erreichen. Gibt es eine analytische Lösung?
- (c) Erzeugen sie eine Liste, die die Kapitalentwicklung über die 10 Jahre bei einem Anfangskapital von 10000 Euro darstellt. Verwenden Sie dazu das eingebaute Prädikat `between/3`

**2. Approximation einer nichtrationalen Zahl** (10 Punkte, Zeitbedarf max. 40 Min.)

Die Eulersche Zahl  $e$  kann durch folgende Reihenzerlegung approximiert werden:

$$e = 1 + \frac{1}{1!} + \frac{1}{2!} + \dots = 1 + \sum_{n=1}^{\infty} \frac{1}{n!}$$

- (a) Definieren Sie unter Verwendung der arithmetischen Auswertungsumgebung ein Prädikat, das  $e$  mit einer vorgegebenen Anzahl von Rekursionsschritten berechnet.

```
?- e(10,Resultat).
Resultat = 2.71828 ;
No
```

Hinweis: Definieren Sie sich hierzu ein (Hilfs-)Prädikat zur Berechnung der Fakultät einer natürlichen Zahl.

- (b) Terminiert ihre Definition aus a) auch bei der Anforderung alternativer Lösungen? Wenn nicht, wie muss Ihre Implementation verändert werden?

*Fortsetzung auf der nächsten Seite*

- (c) Untersuchen Sie Ihre Prädikatsdefinition auf Effizienzreserven. Integrieren Sie die Berechnung der Fakultät so die rekursive Approximation von  $e$ , dass unnötige Doppelberechnungen vermieden werden.
- (d) Ermitteln Sie, wie viele Approximationsschritte erforderlich sind, um  $e$  mit einer Genauigkeit von  $10^{-6}$  zu berechnen.  
Hinweis: Zur Überprüfung der Ergebnisse können Sie die Darstellungsgenauigkeit für Gleitkommazahlen erhöhen, indem Sie

```
?- set_prolog_flag(float_format, '%.18g').
```

am Systemprompt eingeben.

### 3. Rekursive Synthese grafischer Objekte (10 Punkte, max. Zeitbedarf 40 Minuten)

Ergänzen Sie das unvollständige Programm in der Datei `grafik.pl` und erzeugen Sie damit verschiedene Grafiken. Orientieren Sie sich dabei an den Beispielprogrammen auf der P1-Seite unter "Galerie".

Reichen Sie Ihre beste Lösung zusammen mit dem Programm, das sie erzeugt hat, bei Ihrem Übungsgruppenleiter zu Ausstellung in der "Galerie P1" ein. Die besten Einreichungen in den beiden Kategorien

- coolste Grafik bzw.
  - Grafik, die die Idee der Rekursion am besten veranschaulicht
- werden mit einem Preis ausgezeichnet.

Hinweise:

Die Grafikanbindung in Prolog erfolgt objektorientiert: Objekte werden mit dem Prädikat `new(Objektname, Objekttyp)` erzeugt. Objektnamen beginnen immer mit dem Zeichen '@'

Grafisch darstellbare Objekte sind:

gerade Linien: `line(X-Anfangspunkt, Y-Anfangspunkt, X-Endpunkt, Y-Endpunkt)`

Bezier-Kurven: `bezier_curve(Anfangspunkt, Endpunkt, Richtung)` bzw.

`bezier_curve(Anfangspunkt, Endpunkt, Richtung1, Richtung2)`

Rechtecke: `box(Breite, Hoehe)`

Kreise: `circle(Radius)`

Ellipsen: `ellipse(Breite, Hoehe)`

Grafische Objekte haben Eigenschaften, die Ihnen mit `send(Objekt, Eigenschaft)` zugewiesen werden können. Eigenschaften sind u.a.

Füllmuster: `fill-pattern(Colour)`

Texturen: `texture(none|dotted|dashed|dashdot|logdash)`

Farben sind vordefiniert (`red, green, blue, black, white, yellow, ...`) oder können selbst kreiert werden:

Farben: `colour(Name, Rotanteil, Gruenanteil, Blauanteil, rgb)`

Objekte können auf einem Display dargestellt werden, indem bei koordinatenlosen Objekten (`box, circle, ellipse`) dem Display ein Punkt mitgeteilt wird, an dem das Objekt zu platzieren ist

`send(Display, display, Objekt, Punkt)`

Punkte zur Angabe von Koordinaten werden als zweistellige Struktur kodiert:

`point(X-Koordinate, Y-Koordinate)`

Mehr Informationen zu den vordefinierten Objekten erhalten Sie über die eingebaute Hilfsfunktion des Prolog-Systems, z.B.

```
?- help(line).
```

*Fortsetzung auf der nächsten Seite*

4. Listenverarbeitung

(8 Punkte, Zeitbedarf maximal 40 Minuten)

- (a) Unifizieren Sie die folgenden Listen und geben Sie ggf. an, welche Variablenbindungen dabei erzeugt werden.

[a,A]      [B,b]  
[[a,b]]    [H|T]  
[c|R]      [H,H,H]  
[B,B|T]    [[R,m], [n|S], [R,S], [S|R]]

- (b) Das eingebaute Prädikat `last/2` überprüft die Identität des letzten Elementes einer Liste. Reimplementieren sie dieses Prädikat unter Verwendung einer rekursiven Definition als Prädikat `mylast/2` und testen Sie es in verschiedenen Instanziierungsvarianten.

- (c) Modifizieren Sie Ihre Prädikatsdefinition so, dass
- i. das vorletzte Element einer Liste ermittelt wird bzw.
  - ii. als Resultat eine Zweierliste aus dem ersten und dem letzten Element einer Liste zurückgegeben wird.

Beschreiben Sie das Verhalten Ihrer Prädikate in wichtigen Grenzfällen.

Bonus assignment: Thinking about recursion (just for fun)

The fundamental idea of recursion becomes increasingly commonplace in our everyday life. Recently “The Onion”, America’s finest news source, announced two major breakthrough inventions both based on the notion of a recursive relationship, see

<http://www.theonion.com/onion3325/nicarest.html>

and

<http://www.theonion.com/onion3104/newremote.html>.

Compare these two innovations with recursive relationships as discussed in our course. Pay particular attention to the availability of termination criteria in both cases.

*Viel Erfolg!*