WS 2000/2001 E. Fehr

Übungsklausur zu ALGORITHMEN UND PROGRAMMIERUNG I

Abgabe nicht erforderlich

Zum Bestehen dieser Klausur wären 13 Punkte nötig. Besprechung im nächsten Tutorium.

1. **Aufgabe** *7 Punkte*

   Die Anzahl der k-elementigen Teilmengen einer n-elementigen Menge beträgt

   $$C_n^k \;=\; \binom{n}{k} \;=\; \frac{n!}{k!(n-k)!}$$

   (a) Schreiben Sie eine Haskell-Funktion, die den Wert C_n^k für die Argumente n und k ermittelt. Programmieren Sie dabei die Fakultätsfunktion unter Verwendung der Akkumulatortechnik.

   (b) Die Wahrscheinlichkeit $wlotto$, im Lottospiel *6 Richtige* zu haben, wird durch die Anzahl der Treffer, geteilt durch die Anzahl der möglichen Tips ermittelt. Definieren Sie die Größe $wlotto$ als Konstante in Haskell für das Spiel 6 aus 49.

2. **Aufgabe** *8 Punkte*

   In Quotopia herrscht ein strenges Quoten-Gesetz. Gremien mit Entscheidungskompetenz haben grundsätzlich n (n ist ein Vielfaches von 3) Mitglieder, wobei das Verhältnis von Frauen und Männern stets 2:1 ist. Nach den Wahlen zu einem Gremium wird die Liste f der mit wenigstens einer Stimme gewählten Frauen bis zur Länge $\frac{2}{3}n$ und die der Männer m bis zur Länge $\frac{1}{3}n$ in absteigender Reihenfolge (nach entfallenen Stimmen) sortiert und zur Gremienbesetzung so verschmolzen, dass

   i. die Quote 2:1 erfüllt ist
   ii. wenn eine Liste kürzer als $\frac{2}{3}n$ bzw. $\frac{1}{3}n$ ist, das Gremium entsprechend verkleinert wird (Sparmassnahme)
   iii. die Kandidaten auf jeder Liste gemäß ihrer Reihenfolge berücksichtigt werden.

   (a) Schreiben Sie eine Haskell-Funktion, die aus den Listen f und m die Liste g der Mitglieder des neu gewählten Gremiums erzeugt.

   (b) Beweisen Sie die Eigenschaft (i)

bitte wenden

3. **Aufgabe** *5 Punkte*

Die natürlichen Zahlen lassen sich durch den algebraischen Datentyp

$$data \;\; Nat \;\; ::= \;\; Null \;\; | \;\; S \;\; Nat$$

einführen. Alle arithmetischen Operationen lassen sich rekursiv über Nat definieren. Z.B.

$$\begin{array}{ll} plus \;\; :: \;\; Nat \;\longrightarrow\; Nat \;\longrightarrow\; Nat & \\ plus \;\; Null \;\; n & = \;\; n \\ plus \;\; (S\, m) \;\; n & = \;\; S \;\; (plus \;\; m \;\; n) \end{array}$$

(a) Definieren Sie in analoger Weise: die Subtraktion $minus$ mit $minus \;\; m \;\; n \;\; = \;\; Null$, falls m kleiner n gilt.

(b) Beweisen Sie, dass für alle $m, n \;\; :: \;\; Nat$ gilt: $minus \;\; (plus \;\; m \;\; n) \;\; n \;\; = \;\; m$. Begründen Sie Ihre Beweisschritte genau.

4. **Aufgabe** *6 Punkte*

Programmieren Sie eine Haskell-Funktion, die erkennt, ob ein Text t einen Text s enthält. Betrachten Sie t und s als Folge von $char$