Prof. Dr. Manfred Schmidt-Schauß
Künstliche Intelligenz/Softwaretechnologie

Fachbereich Informatik (20)
Johann Wolfgang Goethe-Universität Frankfurt am Main

Praktische Informatik 1:

Wintersemester 1997/98

Übungsblatt Nr. 7

Abgabe: Freitag 23. Januar 1998 vor der Vorlesung

Bonus-Aufgabe 1 (10 Punkte)

Zeigen Sie, daß die folgende Implementierung von Nat korrekt ist:

```
null = []
succ x = 5:x
pred (_:x) = x
```

Aufgabe 2 (20 Punkte)

1. Zeigen Sie, daß

   ```
   null = 0
   succ x = x+1
   pred x = x-1
   neg x = -x
   ```

   für `Int` keine korrekte Hugs-Implementierung des Datentyps GZ ist.

2. (a) Spezifizieren Sie axiomatisch einen Datentyp Int mit den Operationen plus, minus und negiere, der Zahlem von $-2^{31} \dots 2^{31}-1$ darstellen kann und bei einem Unterlauf bzw. Überlauf oben bzw. unten weiterzählt.

   (b) Geben Sie eine Hugs-Implementierung, `MyInt` von Int mit Hilfe des Hugs-Typs `Integer` an.

3. Verändern Sie die axiomatische Semantik von GZ so, daß die Hugs-Implementierung mit `MyInt` für die veränderte Semantik korrekt ist. Verwenden Sie Abkürzungen für Zahlen.

4. Zeigen Sie, daß dann die Hugs-Implementierung korrekt ist.

Aufgabe 3 (20 Punkte)

1. Geben Sie die Funktionalität eines Datentyps endliche Menge als axiomatische Semantik an, der Test auf Enthaltensein, Bildung von Schnitt, Vereinigung und Differenz und Erzeugen einer Menge mit einem Element zur Verfügung stellt.

2. Geben Sie eine Implementierung an und
3. zeigen Sie, daß Ihre Implementierung des Tests auf Enthaltensein korrekt ist.

Aufgabe 4 (15 Punkte)

Benutzen Sie folgende Datenstruktur für Bäume mit mehreren Töchtern pro Knoten:

```
data NBaum a = NBlatt a | NKnoten [NBaum a]
```

1. Schreiben Sie hierfür die Funktionen `nb_rand`, `nb_map` und `foldnb` analog zu `b_rand`, `b_map` und `foldbt`.
2. Schreiben Sie eine Funktion `nb_generate xs n`, die die Elemente der Liste `xs` ungefähr gleichmäßig auf die Blätter verteilt, so daß jeder Knoten höchstens `n` Töchter hat und nur logarithmisch mit der Anzahl der Elemente in `xs` zunimmt.
3. Testen Sie Ihre Funktionen in sinnvoller Weise.