7. Übungsblatt zu Programmierung

Prof. Gert Smolka, Thorsten Brunklaus
www.ps.uni-sb.de/courses/prog-ws00/

Abgabe: 15. Dezember 2000 in der Vorlesungspause (auf Papier)

Allgemeine Hinweise: Die Übungsblätter sollen in Zweiergruppen bearbeitet werden. Die Lösungen geben Sie bitte Freitag bis zur Vorlesungspause auf Papier ab. Jede Gruppe soll nur eine Lösung einreichen, versehen mit den Namen und den Matrikelnummern der Gruppenmitglieder, sowie der Übungsgruppennummer.

Aufgabe 7.1: Rekursionsbäume und Laufzeit(2+2+2) Sei die folgende Prozedur gegeben:

```
fun p n = if n< 4 then () else (p(n-1) ; p(n div 2))
```

(a) Zeichnen Sie den Rekursionsbaum für den Aufruf `p 8`.

(b) Geben Sie die Laufzeit des Aufrufs `p 8` an.

(c) Schreiben Sie eine Prozedur `phi: int->int`, die die Laufzeit von `p` berechnet.

Aufgabe 7.2: Größter gemeinsamer Teiler(1+2+2+3+3+2+3) Sei die folgende Prozedur gegeben:

```
fun gcd(x,y) = if x=y then x
               else if x<y then gcd(x, y-x) else gcd(x-y, y)
```

(a) Ist `gcd` iterativ?

(b) Zeichnen Sie den Rekursionsbaum für den Aufruf `gcd(216,60)`.

(c) Schreiben Sie eine iterative Prozedur `phi: int*int->int`, die die Laufzeit von `gcd` berechnet.

(d) Zeigen Sie, dass `gcd` für alle Argumente $(x, y) \in \mathbb{N}^+ \times \mathbb{N}^+$ terminiert. Geben Sie dazu die Terminierungsbedingungen und eine passende wohlfundierte Relation an.

(e) Sei ϕ die Laufzeit von `gcd`. Beweisen Sie durch Induktion über $\max\{x, y\}$:

$$\forall x, y \in \mathbb{N}^+ : \ \phi(x, y) \leq \max\{x, y\}$$

(f) Geben Sie für alle $x \in \mathbb{N}^+$ ein $y \in \mathbb{N}^+$ an, sodass $\phi(x, y) = \max\{x, y\}$.

(g) Beweisen Sie, dass `gcd` für alle Argumente $(x, y) \in \mathbb{N}^+ \times \mathbb{N}^+$ den größten gemeinsamen Teiler von x und y berechnet. Nehmen Sie dazu an, dass `gcd` für diese Argumente terminiert (siehe Teil (4)).

Aufgabe 7.3: Iteratives Berechnen von balancierten Binärbäumen(2+2) Gegeben sei der Typkonstruktor `tree` für Binärbäume aus Abschnitt 6.5.

(a) Schreiben Sie eine Prozedur

```
tree : int -> unit tree
```

die für $n \in \mathbb{N}$ einen balancierten Binärbaum der Tiefe n mit linearer Laufzeit berechnet. Dabei soll nur iterative Rekursion verwendet werden. Verwenden Sie eine Hilfsprozedur `tree'` mit einem Akkumulatorargument.

(b) Schreiben Sie eine Prozedur

```
tree : int -> int tree
```

die für $n \in \mathbb{N}$ einen balancierten Binärbaum der Tiefe n berechnet, dessen Knoten mit der Tiefe des jeweils erreichbaren Teilbaums markiert sind. Beispielsweise soll

```
tree 2 = N(2, N(1, L 0, L 0), N(1, L 0, L 0))
```

gelten. Dabei soll nur iterative Rekursion verwendet werden und die Laufzeit soll linear sein. Verwenden Sie eine Hilfsprozedur `tree'` mit zwei Akkumulatorargumenten.

Aufgabe 7.4: Induktion über Listen(3+3+3+3+3) Beweisen Sie die Teile (2)-(6) von Proposition 7.6.1.Dabei dürfen Sie Teil (1) der Proposition verwenden.

Aufgabe 7.5: Iteratives Length(3+3+3) Sei X eine Menge.

(a) Definieren Sie mit iterativer Rekursion eine Funktion

$$len' \in \mathcal{L}(X) \times \mathbb{Z} \to \mathbb{Z}$$

für die gilt:

$$\forall xs \in \mathcal{L}(X)\ \forall n \in \mathbb{Z}:\ len'(xs, n) = |xs| + n$$

(b) Beweisen Sie mit Listeninduktion, dass ihre Funktion diese Gleichung erfüllt.

(c) Schreiben Sie eine Prozedur `len: 'a list -> int`, die die Länge eine Liste mithilfe einer iterativen Hilfsprozedur berechnet. Die Hilfsprozedur soll ihre Funktion *len'* berechnen. Geben Sie exakte Laufzeit ihrer Prozedur `len` für beliebige Listen `xs` an.