Musterlösung zum 6. Aufgabenblatt vom Montag, den 25. Mai 2009 zur Vorlesung

Informatik B

(Frank Hoffmann)

von Thore Kübart

1. **Registermaschinen** (2+2 Punkte) Schreiben Sie für eine Registermaschine zwei Programme, die die Fakultätsfunktion bzw den diskreten Zweierlogarithmus (also $\lfloor \log_2 n \rfloor$) einer positiven natürlichen Zahl berechnen.

   Lösung:

   `Fakultätsfunktion:`

   Mit dem `IF-GOTO`-Befehl kann eine `DO-WHILE`-Schleife realisiert werden. Ein möglicher Pseudocode, um in `result` das Ergebnis zu speichern, wenn in `n` die Eingabe ≥ 1 gespeichert ist, wäre also:

```
result:=1
DO
  result:=result*n
  n:=n-1
WHILE n>1
```

   Dieser übersetzt sich wie folgt in die Sprache der Registermaschine, wobei in `c(1)` die Eingabe, in `c(2)` das Ergebnis gespeichert sei:

```
1   CLOAD 1
2   STORE 2
3   LOAD 2
4   MUL 1
5   STORE 2
6   LOAD 1
7   CSUB 1
8   STORE 1
9   IF c(0) > 1 GOTO 3
```

   `diskrete Zweierlogarithmus:`

   Ist $n \in \left[2^k, 2^{k+1}\right) \cap \mathbb{N}$ mit $k \in \mathbb{N}$, so muss man genau k-mal, also $\lfloor \log_2 n \rfloor$-mal auf n die Operation $\lfloor n/2 \rfloor$ (das entspricht der Operation `CDIV 2`) anwenden, um $n = 1$ zu erhalten! Mit dieser Überlegung ergibt sich der folgende Pseudocode:

```
result:=0
DO
  result:=result+1
```

```
  n:=round(n\2)
WHILE n>0
result:=result-1
```

Für den Code der Registermaschine sei in `c(1)` die Eingabe gespeichert, alle anderen Speicherzellen erhalten zu Beginn den Wert 0. In `c(2)` wird wieder das Ergebnis gespeichert.

```
1   LOAD 2
2   CADD 1
3   STORE 2
4   LOAD 1
5   CDIV 2
6   STORE 1
7   IF c(0) > 0 GOTO 1
8   LOAD 2
9   CSUB 1
```

2. **O–Notation I** (8 Punkte)
Beweisen oder widerlegen Sie folgende Aussagen. Dabei sind $f(n)$ und $g(n)$ beliebige positive Funktionen mit natürlichen Zahlen als Argumente.

   (b) $f(n) = \Theta(f(2n))$
   Lösung:
   Die Aussage ist falsch. Ein Gegenbeispiel liefert die Funktion $f(n) = 2^n$, da dann $f(n) \neq \Omega(f(2n))$ gilt. Für Gleicheit müsste es ein $c > 0$ geben, so dass für fast alle $n \in \mathbb{N}$ (d.h.: bis auf endlich viele n) gilt: $2^n \geq c * 2^{2n} = (c * 2^n) * 2^n$. Dieses $c > 0$ kann es nicht geben, da immer $(c * 2^n) > 1$ für fast alle $n \in \mathbb{N}$ gilt.

   (c) $f(n) + \omega(f(n)) = \Theta(f(n))$
   Lösung:
   Wieder falsch. Denn betrachte $f(n) = n$, dann ist $n^2 = \omega(n)$ und $n + n^2 \neq \Theta(n)$.

3. **O–Notation II** (2 Punkte) Ist die folgende Argumentation korrekt?
Wir beweisen $\Sigma_{k=1}^{n} k^3 = O(n^3)$ durch Induktion.
Induktionsverankerung Für $n = 1$ gilt $\Sigma_{k=1}^{n} k^3 = 1 = O(n^3)$.
Induktionsannahme $\Sigma_{k=1}^{n} k^3 = O(n^3)$.
Induktionsschritt $\Sigma_{k=1}^{n+1} k^3 = (\Sigma_{k=1}^{n} k^3) + (n+1)^3 = O(n^3) + O(n^3) = O(n^3)$.
Lösung:

   Die Argumentation kann natürlich nicht korrekt sein, da die Behauptung falsch ist. Der Fehler der Argumentation liegt schon in der ersten Zeile. Es wird hier versucht eine Aussage der Form $f(n) = O(g(n))$ mit Induktion über n zu zeigen. Das führt dazu, dass für jeden einzelnen Funktionswert gezeigt wird, dass er, eine Konstante, in $O(g(n))$ liegt. Damit ist natürlich keinesfalls $f(n) = O(g(n))$ gezeigt.

   Auch sollte klar sein, dass die im Induktionsbeweis auftretenden n unterschiedliche Rollen haben. Im Induktionsschritt wird zum Beispiel $(n+1)^3 = O(n^3)$ „ausgenutzt“. An dieser Stelle würde zum Beispiel auch „$(n+1)^3 = O(n)$“ gelten, denn das n auf

der linken Seite bezeichnet eine konkrete natürliche Zahl, $(n+1)^3$ ist hier also eine Konstante, während das n auf der rechten Seite eine Variable bezeichnet mit deren Hilfe eine Funktion beschrieben wird.