technische universität dortmund

SoSe 2017

C. Sohler
A. Krivošija, A. Rey, H. Sandvoß

DAP2 – Heimübung 3

Ausgabedatum: 28. 4. 17 — Abgabedatum: Fr. 5. 5. 17 (Mo. 8. 5. für Gruppen 27–32) 12 Uhr

Abgabe:

Schreiben Sie unbedingt immer Ihren vollständigen Namen, Ihre Matrikelnummer und Ihre Gruppennummer auf Ihre Abgaben! Beweise sind nur dort notwendig, wo explizit danach gefragt wird. Eine Begründung der Antwort wird allerdings *immer* verlangt.

Aufgabe 3.1 (5 Punkte): (Schleifeninvariante und Korrektheitsbeweis)

Betrachten Sie das folgende Programme, das als Eingabe eine natürliche Zahl $n \in \mathbb{N}$ erhält.

BerechneWert1(n):
if $n = 0$ **then**
    return 0
$a \leftarrow 1$
$b \leftarrow 1$
for $i = 1$ **to** $n - 1$ **do**
    $c \leftarrow b - i$
    $b \leftarrow c + a$
    $a \leftarrow c + 2i + 1$
return b

Untersuchen Sie die Funktionsweise und Korrektheit dieses Programms, indem Sie wie folgt vorgehen:

a) Stellen Sie eine Behauptung auf, welchen Wert das Programm *BerechneWert1* für eine Eingabe n ausgibt.

b) Formulieren Sie zwei Schleifeninvarianten, die zu Beginn jeder Iteration der obigen Schleife im Programm *BerechneWert1* für die Variablen a und b gelten, und beweisen Sie diese mittels vollständiger Induktion.

c) Verwenden Sie dann diese Schleifeninvarianten, um zu zeigen, dass die Behauptung in Aufgabenteil (a) korrekt ist.

Hinweis: Sie können gegebenenfalls auf Funktionsdefinitionen aus vorherigen Übungsaufgaben zurückgreifen.

Lösung:

a) Behauptung: Das Programm *BerechneWert1* gibt für eine Eingabe n die n-te Fibonacci-Zahl $f(n)$ aus.

b) Die gesuchten Schleifeninvarianten für das Programm *BerechneWert1* lauten:

$A_1(i)$*: Zu Beginn der i-ten Iteration der For-Schleife gilt* $a = f(i-1) + i$;
$A_2(i)$*: Zu Beginn der i-ten Iteration der For-Schleife gilt* $b = f(i)$.

(Anmerkung: Die Fibonacci-Zahlen sind spätestens aus dem Präsenzblatt 2 bekannt.)

Wir beweisen beide Aussagen mittels vollständiger Induktion über i.

(I.A.) Wenn $i = 1$ ist, betreten wir den Schleifenrumpf zum ersten Mal. Da a in Zeile 3 mit 1 initialisiert wird, gilt $a = 1 = f(0)+1$. Somit ist Invariante A_1 für $i = 1$ erfüllt. Da b in Zeile 4 mit 1 initialisiert wird, gilt $b = 1 = f(1)$, also auch Invariante A_2 für $i = 1$.

(I.V.) Für ein festes $i_0 < n$ gelten beide Invarianten $A_1(i_0)$ und $A_2(i_0)$, d. h., zu Beginn des i_0-ten Schleifendurchlaufs seien $a = f(i_0 - 1) + i$ und $b = f(i_0)$.

(I.S.) Sei $i = i_0 + 1$. Zu zeigen sind $A_1(i)$ und $A_2(i)$, also dass nach dem i_0-ten Schleifendurchlauf (was dem Zeitpunkt vor dem i-ten Durchlauf oder für $i_0 = n - 1$ dem Schleifenaustritt entspricht), $a = f(i-1) + i$ und $b = f(i)$ gelten.

Nach Induktionsvoraussetzung sind zu Beginn des i_0-ten Durchlaufs $a = f(i_0-1)+1$ und $b = f(i_0)$. Während des i_0-ten Schleifendurchlaufs werden dann in c auf $b - i_0 = f(i_0) - i_0$ (Zeile 6), b auf $c + a = f(i_0) - i_0 + f(i_0 - 1) + i = f_(i_0 + 1)$ (Zeile 7) und a auf $c + 2i + 1 = f(i_0) - i_0 + 2i_0 + 1 = f(i_0) + i_0 + 1$ (Zeile 8) gesetzt.

Somit gelten für $i = i_0+1$ mit $a = f(i-1)+i$ und $b = f_(i_0+1) = f(i)$ die Invarianten auch.

Nach dem Prinzip der vollständigen Induktion gilt die Behauptung der Schleifeninvarianten zu Beginn jeder Schleifeniteration und vor dem Schleifenaustritt.

c) Nach Teilaufgabe b) gelten die Invarianten $A_1(i)$ und $A_2(i)$ vor dem Schleifenaustritt, d. h. nach dem Schleifendurchlauf $n - 1$, also für $i = n$. Damit enthält die Variable b den Wert $f(n)$, der in Zeile 9 zurückgegeben wird. $\square$

Aufgabe 3.2 (5 Punkte): (Korrektheitsbeweis mit Rekursion)

Betrachten Sie das folgende rekursive Programm, das eine natürliche Zahl n erhält.

```
   BerechneWert2(n):
1  if n = 0 then
2      return 0
3  if n = 1 then
4      return 1
5  if n = 2 then
6      return 1
7  return 2 · BerechneWert2(n − 1) − BerechneWert2(n − 3)
```

Stellen Sie eine Behauptung auf, was dieses Programm für eine Eingabe n ausgibt, und beweisen Sie die Korrektheit Ihrer Behauptung mittels vollständiger Induktion.

Lösung: *Behauptung:* Das Programm *BerechneWert2* berechnet ebenfalls für eine Eingabe n die n-te Fibonacci-Zahl $f(n)$.

Beweis: Wir zeigen dies mittels erweiterter vollständige Induktion über n.

(I.A.) Wenn $n = 0$ ist, wird $0 = f(0)$ in Zeile 2 zurückgegeben; wenn $n = 1$ ist, wird $1 = f(1)$ in Zeile 4 zurückgegeben; wenn $n = 2$ ist, wird $1 = f(2)$ in Zeile 6 zurückgegeben.

(I.V.) Sei $n_0 \geq 2$ eine beliebige natürliche Zahl. Für alle Eingaben $n \leq n_0$ gebe das Programm jeweils $f(n)$ zurück.

(I.S.) Betrachte den Aufruf *BerechneWert2*(n) für $n = n_0 + 1$. Da $n > 2$ ist die Bedingung in den If-Abfragen in Zeilen 1, 3 und 5 falsch, sodass direkt Zeile 7 ausgeführt wird. Nach der Induktionsvoraussetzung gilt, dass *BerechneWert2*($n-1$) mit $n - 1 = n_0$den Wert $f(n_0)$ und *BerechneWert2*($n-3$) mit $n-3 = n_0 - 2 \leq n_0$ den Wert $f(n_0 - 2)$ zurückgibt. D. h., das Programm gibt den Wert

$$\begin{aligned} 2 \cdot f(n_0) - f(n_0 - 2) &= f(n_0) + f(n_0) - f(n_0 - 2) \\ &= f(n_0) + f(n_0 - 1) + f(n_0 - 2) - f(n_0 - 2) \\ &= f(n_0 + 1) = f(n) \end{aligned}$$

zurück.

Nach dem Prinzip der vollständigen Induktion gilt die Behauptung für alle $n \in \mathbb{N}$. $\square$

Anmerkung: Das erste Programm aus Aufgabe 3.1 entspricht im Wesentlichen (die Addition und anschließende Subtraktion von i ist überflüssig) dem interativen Programm zur Berechnung einer Fibonacci-Zahl, welches in $\mathcal{O}(n)$ arbeitet. Das zweite Programm aus Aufgabe 3.2 entspricht im Wesentlichen (bis auf die etwas umständliche Formulierung des Aufrufs in der letzten Zeile) der rekursiven Variante. Diese läuft in $\Theta(\phi^n)$) für den goldenen Schnitt $\phi = \frac{1+\sqrt{5}}{2} \approx 1,61803$.