Übungsblatt 6

Adrian Schollmeyer

Aufgabe 1

(a)

Nicht lösbar. Der Anfang mit Index 1 ist unmöglich, da α_1 mit `0` beginnt, jedoch nicht β_1.

a_2 zusammen mit β_2 erfordert, dass nach jedem Einsetzen von α_2 mindestens zweimal β_2 eingesetzt werden muss, was dazu führt, dass die Kette unendlich lang würde.

Anzufangen mit α_3 erfordert danach β_3, was wiederum entweder β_2 erfordert (was wie oben in einer unendlichen Kette enden würde) oder β_3, was analog dazu auch in einer unendlichen Kette enden würde.

(b)

Nicht lösbar. In jedem Fall ist man gezwungen, mit α_1 anzufangen, wodurch im nächsten Schritt zwingend β_2 eingesetzt werden muss, was wiederum dazu führt, dass in der ersten Kette α_2 (was inhaltsgleich mit α_1 ist) eingefügt wird, wodurch β_1 und darauffolgend β_2 eingefügt werden muss. Dadurch entsteht eine unendliche Kette

(c)

Lösbar: $(5, 3, 3, 2, 5)$

- `10110.1.1.0.10110`
- `10.110.110.101.10`

Aufgabe 2

(a)

Fall 1:

$$\text{Vorraussetzung: } X \geq 0 \wedge Y \geq 0 \tag{1}$$
$$\implies X = X + Y = X + |Y| \quad \forall X \in \mathbb{Z}, Y \in \mathbb{Z}_{\geq 0} \tag{2}$$
$$\text{Nachbedingung: } X \geq 0 \text{ gilt } \forall X \in \mathbb{Z}, Y \in \mathbb{Z}_{\geq 0} \quad \checkmark \tag{3}$$

Fall 2:

$$\text{Vorraussetzung: } X \geq 0 \wedge Y < 0 \tag{4}$$
$$\implies X = X - Y = X + |Y| \quad \forall X \in \mathbb{Z}, Y \in \mathbb{Z}_{<0} \tag{5}$$
$$\text{Nachbedingung: } X \geq 0 \text{ gilt } \forall X \in \mathbb{Z}, Y \in \mathbb{Z}_{<0} \quad \checkmark \tag{6}$$

Daher funktioniert der Algorithmus $\forall X, Y \in \mathbb{Z}$. $\square$

(b)

- Schleifeninvariante: $P = (X \geq 0 \wedge Y \geq 0)$
- $B = X \neq 0$

$$\textit{Vorbedingung} \implies P \text{ wahr}, \tag{7}$$
$$\text{denn } \textit{Vorbedingung} \equiv P \tag{8}$$

Daher gilt P immer auch vor dem Eintritt in die Schleife.

Schleifeninvariante bei Eintritt in die Schleife:

$$P \wedge B \equiv X > 0 \wedge Y \geq 0 \text{ nach Vorbedingung wahr, falls } B \text{ wahr} \quad \checkmark \tag{9}$$

Erhaltung von P im Schleifenrumpf:

$$Y := Y + 1; X := X - 1 \tag{10}$$
$$\big(X' \geq 0 \wedge Y' \geq 0\big) \equiv (X - 1 \geq 0 \wedge Y + 1 \geq 0) \text{ wahr } \forall X > 0, Y \geq 0 \quad \checkmark \tag{11}$$

Nachbedingung:

$$P \equiv (X \geq 0 \wedge Y \geq 0) \implies Y \geq 0 \text{ wahr} \quad \checkmark \tag{12}$$

Damit ist die Korrektheit des Algorithmus gezeigt. $\square$

(c)

Da der Algorithmus unabhängig der Eingabe nie terminiert (aufgrund des `while true` und da es keine sonstige Abbruchanweisung innerhalb des Schleifenrumpfes gibt), ist die Aussage korrekt.

Aufgabe 3

- $P = (N \geq 0 \wedge C \geq 0 \wedge C \leq N \wedge S \geq 0)$
- $B = (C \neq N)$

$$\begin{aligned}
\textit{Vorbedingung} &\implies P && (1)\\
N \geq 0 &\implies (N \geq 0 \wedge C \geq 0 \wedge C \leq N \wedge S \geq 0) && (2)\\
N \geq 0 &\implies (N \geq 0 \wedge 0 \geq 0 \wedge 0 \leq N \wedge 0 \geq 0) && (3)\\
N \geq 0 &\implies (N \geq 0 \wedge 0 \leq N) \text{ wahr } \checkmark && (4)
\end{aligned}$$

Bedingung beim Schleifeneintritt:

$$P \wedge B \equiv (N \geq 0 \wedge C \geq 0 \wedge C < N \wedge S \geq 0) \tag{5}$$

Nach Vorbedingung wahr, falls $C \neq N$ ✓

Korrektheit von P im Schleifenrumpf:

$$\begin{aligned}
P &\equiv \big(N \geq 0 \wedge C' + 1 \geq 0 \wedge C' + 1 \leq N \wedge S' + C \geq 0\big) && (6)\\
&\equiv \big(\text{wahr} \wedge \text{wahr} \wedge C' + 1 \leq N \wedge S' + C \geq 0\big) \text{ wahr} && (7)\\
\text{denn } C' < N &\text{ nach } B && (8)\\
\text{und } S' + C \geq 0 &\text{ wahr, denn } S' > 0 \wedge C > 0 \ \checkmark && (9)
\end{aligned}$$

Daher ist P auch im Schleifenrumpf gültig.

Gültigkeit der Nachbedingung:

$$P \implies \left(S = N \cdot \frac{N+1}{2}\right) \tag{10}$$

Aufgabe 4

(a) $VOR \equiv N \geq 0$

(b) $NACH \equiv A = \dfrac{\left(\frac{1+\sqrt{5}}{2}\right)^{n-1} - \left(-\frac{1+\sqrt{5}}{2}\right)^{-n-1}}{\sqrt{5}}$

Aufgabe 5

(a)

Listing 1: AsymMain.java

```java
package me.adrian;

public class AsymMain
{
    public static int xSwitch(double a, double b)
    {
        if (a < 1) {
            throw new ArithmeticException("a must be greater than 1");
        }
        if (b < 1) {
            throw new ArithmeticException("b must be greater than 1");
        }
        int x = 2;
        while (true) {
            if (Math.pow(x, a) < Math.pow(b, x)) {
                return x;
            }
            x++;
        }
    }

    public static void main(String[] args)
    {
        double a = Double.parseDouble(args[0]);
        double b = Double.parseDouble(args[1]);
        System.out.println(Integer.toString(xSwitch(a, b)));
    }
}
```

(b)

$$x \geq 1499$$