

1.

```

z := 0;
r := x;
while y < r do
  r := r - y;
  z := z + 1
end

```

Folglich ist die durch das obige Stück Programmcode berechnete Funktion

$$\begin{pmatrix} r \\ z \end{pmatrix} := \begin{pmatrix} x \bmod y \\ \lfloor \frac{x}{y} \rfloor \end{pmatrix}$$

2.

- *Induktionsanfang*

$\langle c_1, e_1, s \rangle \triangleright^n \langle c', e', s' \rangle \Rightarrow \langle c_1 : c_2, e_1 e_2, s \rangle \triangleright^n \langle c' : c_2, e' : e_2, s' \rangle$, für $n := 1$
ist soweit klar, da wir hier genau einen Ableitungsschritt durchführen.

- *Induktionsvoraussetzung*

$$n := k, \quad k \in \mathbb{N}^+$$

- *Induktionsbehauptung*

Obige Implikation gilt auch für $n = k + 1$.

- *Induktionsschluss*

$$\langle c_1, e_1, s \rangle \triangleright^{k+1} \langle c', e', s' \rangle \Rightarrow \langle c_1 : c_2, e_1 e_2, s \rangle \triangleright^{k+1} \langle c' : c_2, e' : e_2, s' \rangle$$

Wir können dies nun nach dem Induktionsanfang aufspalten in einmal k Schritte und einmal einen Schritt. Führen wir einen Schritt aus [nach Induktionsanfang], so haben wir

$$\langle c', e', s' \rangle \triangleright^k \langle c'', e'', s'' \rangle \Rightarrow \langle c' : c_2, e' e_2, s' \rangle \triangleright^k \langle c'' : c_2, e'' : e_2, s'' \rangle,$$

was nach Induktionsbehauptung und -voraussetzung gilt.

3.

$S = \text{while } x \neq 0 \text{ do } x := x - 1 \text{ end}$

$$F_S(g)(s) = \begin{cases} g \circ \mathcal{S}_{\text{ds}} \llbracket x := x - 1 \rrbracket & \text{falls } s(x) \neq 0 \\ s & \text{falls } s(x) = 0 \end{cases}$$

Bzw.:

$$F_S(g)(s) = \begin{cases} g \circ (s[x \rightarrow s(x) - 1]) & \text{falls } s(x) \neq 0 \\ s & \text{falls } s(x) = 0 \end{cases}$$

Nun prüfen wir, ob die Funktionen $g_i(s)$, ($i \in \{1, 2, 3, 4, 5\}$) tatsächlich Fixpunkte von F_S sind:

- $g_1(s) = \text{undef}$

Für $s(x) > 0$ ist $F_S(g_1)(s) = s[x \rightarrow 0] \neq g_1(s) = \text{undef}$. Folglich ist g_1 kein Fixpunkt.

- $g_2(s) = \begin{cases} s[x \rightarrow 0], & \text{falls } s(x) \geq 0 \\ \text{undef}, & \text{falls } s(x) < 0 \end{cases}$

Falls $s(x) \geq 0$, so ist $F(g_2)(s) = s[x \rightarrow 0] = g_2(s)$.

Falls $s(x) < 0$, so ist $F(g_2)(s) = \text{undef} = g_2(s)$. Folglich ist g_2 ein Fixpunkt von F_S .

- $g_3(s) = \begin{cases} s[x \rightarrow 0], & \text{falls } s(x) \geq 0 \\ s, & \text{falls } s(x) < 0 \end{cases}$
 Falls $s(x) \geq 0$, so ist $F(g_3)(s) = s[x \rightarrow 0] = g_3(s)$.
 Falls $s(x) < 0$, so ist $F(g_3)(s) = s[x \rightarrow s(x) - 1] \neq g_3(s)$. Damit ist g_3 kein Fixpunkt.
- $g_4(s) = s[x \rightarrow 0]$
 Falls $s(x) \geq 0$, so ist $F_S(g_4)(s) = s[x \rightarrow 0] = g_4(s)$.
 Falls $s(x) < 0$, so ist $F_S(g_4)(s) = s[x \rightarrow 0] = g_4(s)$. Damit ist g_4 ein Fixpunkt.
- $g_5(s) = s$
 Falls $s(x) \geq 0$, so ist $F_S(g_5)(s) = s[x \rightarrow s(x) - 1] \neq g_5(s) = s$. Folglich ist g_5 kein Fixpunkt.

4.

$S := \text{while } x < 1 \text{ do } y := y * x; x := x - 1 \text{ end}$

$$F_S(g)(s) = \begin{cases} g \circ \mathcal{S}_{\text{ds}} \llbracket y := y * x; x := x - 1 \rrbracket & \text{falls } s(x) \neq 1 \\ s & \text{falls } s(x) = 1 \end{cases}$$

Bzw.:

$$F_S(g)(s) = \begin{cases} g \circ (s[y \rightarrow s(y) * s(x)][x \rightarrow s(x) - 1]) & \text{falls } s(x) \neq 1 \\ s & \text{falls } s(x) = 1 \end{cases}$$

Wir können analog zu einigen Beispielen aus der vorangegangenen Aufgabe die folgenden Fixpunkte konstruieren:

$$g_1 := \begin{cases} s[x \rightarrow 1, y \rightarrow 0] & \text{falls } s(x) \neq 1 \\ s & \text{falls } s(x) = 1 \end{cases}$$

sowie

$$g_2 := \begin{cases} \text{undef}, & \text{falls } s(x) < 1 \\ s[x \rightarrow 1, y \rightarrow y * s(x)!] & \text{falls } s(x) \geq 1 \end{cases}$$