4. Aufgabenblatt

(Besprechung in den Tutorien 14.11.2022–18.11.2022)

Aufgabe 1. Primitiv-rekursive Funktionen

Zeigen Sie, dass folgende Funktionen primitiv-rekursiv sind.

1. $g_1 : \mathbb{N}^2 \to \mathbb{N}$ mit $g_1(x, y) := y^x$.
2. $g_2 : \mathbb{N} \to \mathbb{N}$ mit $g_2(n) := \max\{i \in \mathbb{N} \mid i^2 \leq n\}$.

——————Lösungsskizze——————

Im Folgenden sei $c_i^k : \mathbb{N}^k \to \mathbb{N}$ die k-stellige konstante Funktion mit $c_i^k(n_1, \ldots, n_k) = i$.

1. Sei mult die primitiv-rekursive Multiplikationsfunktion (siehe VL).

   Dann lässt sich g_1 wie folgt primitiv-rekursiv definieren.

$$\begin{aligned} g_1(0, y) &= c_1^1(y) = 1 \\ g_1(x+1, y) &= \text{mult} \circ (\pi_2^3, \pi_3^3)(x, g_1(x, y), y) \\ &= \text{mult}(g_1(x, y), y) \end{aligned}$$

2. Wir beobachten, dass $g_2(0) = 0$ ist. Nehmen wir nun an, dass $g_2(n) = \max\{i \in \mathbb{N} \mid i^2 \leq n\} = j$. Wenn nun gilt, dass $(j+1)^2 \leq n+1$, dann ist $g_2(n+1) = j+1 = g_2(n)+1$. Andernfalls ist $g_2(n+1) = j = g_2(n)$. Also gilt:

$$g_2(n+1) = \begin{cases} g_2(n) + 1, & \text{falls } (g_2(n)+1)^2 \leq n+1 \\ g_2(n), & \text{sonst.} \end{cases}$$

   Wir definieren g_2 primitiv-rekursiv wie folgt:

$$\begin{aligned} g_2(0) &= 0 \\ g_2(n+1) &= h(n, g_2(n)), \end{aligned}$$

   wobei $h \colon \mathbb{N}^2 \to \mathbb{N}$ eine Hilfsfunktion ist mit

$$h(n, m) = \begin{cases} m + 1, & \text{falls } (m+1)^2 \leq n+1 \\ m, & \text{sonst.} \end{cases}$$

   Die Funktion h lässt sich primitiv-rekursiv (als Komposition) wie folgt definieren

$$\begin{aligned} h(n, m) &= \text{add} \circ (\text{le} \circ (g_1 \circ (c_2^2, \text{succ} \circ \pi_2^2), \text{succ} \circ \pi_1^2), \pi_2^2)(n, m) \\ &= \text{add}(\text{le}(g_1(2, \text{succ}(m)), \text{succ}(n)), m) \end{aligned}$$

   Die Funktion add ist primitiv-rekursiv (siehe VL). Die *lower-equal* Funktion $\text{le} : \mathbb{N}^2 \to \{0, 1\}$ ist hierbei definiert als

$$\text{le}(x, y) = \begin{cases} 1, & \text{falls } x \leq y \\ 0, & \text{sonst} \end{cases}$$

   und kann durch folgende Hilfsfunktionen primitiv-rekursiv definiert werden:

- Ist Null?

$$N(0) = 1$$
$$N(n+1) = c_0^2(n, N(n)) = 0$$

- (modifizierte) Vorgängerfunktion

$$\text{pred}(0) = 0$$
$$\text{pred}(n+1) = \pi_1^2(n, \text{pred}(n)) = n$$

- (modifizierte) Subtraktion, d.h. $\text{modsub}(x, y) = \max\{0, y - x\}$.

$$\begin{aligned}\text{modsub}(0, y) &= \pi_1^1(y) = y \\ \text{modsub}(x+1, y) &= \text{pred} \circ \pi_2^3(x, \text{modsub}(x, y), y) \\ &= \text{pred}(\text{modsub}(x, y))\end{aligned}$$

Nun gilt $x \leq y$ genau dann, wenn $\text{modsub}(y, x) = 0$, also

$$\text{le}(x, y) = N \circ (\text{modsub} \circ (\pi_2^2, \pi_1^2))(x, y) = N(\text{modsub}(y, x)).$$

Aufgabe 2. μ-Rekursion und LOOP-Programme

Sei $\text{sub}\colon \mathbb{N}^2 \to \mathbb{N}$ mit $\text{sub}(x, y) := \max(0, x - y)$ die modifizierte Subtraktionsfunktion und $\text{mult}\colon \mathbb{N}^2 \to \mathbb{N}$ mit $\text{mult}(x, y) := x \cdot y$ die Multiplikationsfunktion. Außerdem sei $g\colon \mathbb{N}^3 \to \mathbb{N}$ definiert als $g(x, y, z) := \text{sub}(\text{mult}(y, z), x)$.

1. Wieviele Argumente hat die Funktion $\mu(g)$?
2. Zeigen Sie, dass $\mu(g)$ primitiv-rekursiv ist.

——————Lösungsskizze——————

1. g ist vom Typ $\mathbb{N}^3 \to \mathbb{N}$, daher hat $\mu(g)$ den Typ $\mathbb{N}^2 \to \mathbb{N}$, also genau 2 Argumente.
2. Es gilt $\mu(g)(y, z) = yz$, weil für alle $y, z \in \mathbb{N}$ gilt:

$$g(yz, y, z) = \text{sub}(\text{mult}(y, z), yz) = \text{sub}(yz, yz) = \max(0, yz - yz) = 0$$

und für alle $n' < yz$ gilt:

$$g(n', y, z) = \text{sub}(\text{mult}(y, z), n') = \max(0, yz - n') > 0.$$

Diese Funktion können wir mit folgendem LOOP-Programm (mit Eingabewerten $x_1 = y$ und $x_2 = z$) berechnen:

LOOP x_1 **DO**
    LOOP x_2 **DO**
        $x_0 := x_0 + 1$
    END
END

Aufgabe 3. Die 91-Funktion

Zeigen Sie, dass folgende Funktion $f\colon \mathbb{N} \to \mathbb{N}$ μ-rekursiv ist:

$$f(n) := \begin{cases} n - 10, & \text{falls } n > 100, \\ f(f(n+11)), & \text{falls } n \leq 100. \end{cases}$$

Sie können hierbei verwenden, dass WHILE-berechenbare Funktionen μ-rekursiv sind.

Zusatzinformationen: McCarthy 91 Funktion (Wikipedia)

——————Lösungsskizze——————

Folgendes WHILE-Programm berechnet f, wobei die Eingabe n in x_1 steht:

```
x2 := x2 + 1;
WHILE x2 ≠ 0 DO
    IF x1 > 100 THEN
        x1 := x1 − 10;
        x2 := x2 − 1
    ELSE
        x1 := x1 + 11;
        x2 := x2 + 1
    END
END
x0 := x1
```

Daraus folgt, dass f WHILE-berechenbar und damit auch μ-rekursiv ist. Die Idee hinter dem Programm ist, dass f wie in der Definition berechnet wird. Dabei wird ein Ausdruck der Form $f(f(\ldots(f(n))\ldots))$ aufgebaut. In der Variable x_2 wird die Schachtelungstiefe der Funktionsaufrufe gespeichert. In x_1 wird n gespeichert. Die Korrektheit dieses Programms zu beweisen ist jedoch etwas umständlich (Induktion über x_2). Deswegen analysieren wir die Funktion f genauer.

Bei genauerer Betrachtung stellt sich heraus, dass folgendes gilt:

$$f(n) = \begin{cases} n - 10, & \text{falls } n > 100, \\ 91, & \text{sonst.} \end{cases} \tag{1}$$

Das impliziert, dass f WHILE-berechenbar und somit μ-rekursiv ist, denn diese Funktion wird von folgendem trivialen WHILE-Programm berechnet:

```
IF x1 > 100 THEN
    x0 := x1 − 10
ELSE
    x0 := x0 + 91
END
```

Es ist also zu zeigen, dass (1) gilt. Falls, $n \geq 101$ ist die Behauptung klar.

Wir müssen zeigen, dass $f(n) = 91$, wenn $n \leq 101$. Wir zeigen dies per Induktion über $k(n) := 101 - n$. Falls $k(n) = 0$, folgt die Behauptung aus dem obigen. Sei also $k(n) > 0$. Sei zunächst $90 \leq n \leq 100$. Dann ist

$$f(n) = f(f(n+11)) = f(n + 11 - 10) = f(n+1).$$

Da $k(n+1) < k(n)$, folgt $f(n+1) = 91$ aus der Induktionsvoraussetzung. Sei nun $n < 90$. Dann ist $f(n) = f(f(n+11)) = f(91) = 91$, wobei die letzten beiden Gleichheiten aus der Induktionsvoraussetzung folgen. Damit ist (1) bewiesen.