Universität Leipzig
Institut für Informatik
Sommersemester 2025

Prof. Dr. Andreas Maletti,
Dr. habil. Karin Quaas,
Fabian Sauer

Aufgaben zur Lehrveranstaltung

Berechenbarkeit

Lösungen zu Serie 4

Übungsaufgabe 4.1 (Ackermann & Co)

Wir betrachten die Ackermann-Funktion $a : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, die wie folgt definiert ist:

$$a(x, y) = \begin{cases} y + 1 & \text{falls } x = 0 \\ a(x-1, 1) & \text{falls } x \neq 0 \text{ und } y = 0 \\ a(x-1, a(x, y-1)) & \text{sonst.} \end{cases}$$

Zeigen Sie, dass für alle $s \in \mathbb{N}$ gilt:

(a) $a(1, s) = s + 2$

(b) $a(2, s) = 2 \cdot s + 3$

LÖSUNG: (a) Per Induktion über s.

Induktionsanfang:

$$\begin{aligned} a(1, 0) &= a(0, 1) \quad \text{(Fall 2 der Definition von } a\text{)} \\ &= 2 \quad \text{(Fall 1 der Definition von } a\text{)} \\ &= 0 + 2 \end{aligned}$$

Induktionsschritt:

$$\begin{aligned} a(1, s+1) &= a(0, a(1, s)) \quad \text{(Fall 3 der Definition von } a\text{)} \\ &= a(1, s) + 1 \quad \text{(Fall 1 der Definition von } a\text{)} \\ &= s + 2 + 1 \quad \text{(Induktionshypothese)} \\ &= (s + 1) + 2 \end{aligned}$$

(b) Per Induktion über s.

Induktionsanfang:

$$\begin{aligned} a(2, 0) &= a(1, 1) \quad \text{(Fall 2 der Definition von } a\text{)} \\ &= 1 + 2 \quad \text{(Aufgabe (3.1.a))} \\ &= 2 \cdot 0 + 3 \end{aligned}$$

Induktionsschritt:

$$\begin{aligned} a(2, s+1) &= a(1, a(2, s)) \quad \text{(Fall 3 der Definition von } a\text{)} \\ &= a(2, s) + 2 \quad \text{(Aufgabe (3.1.a))} \\ &= 2s + 3 + 2 \quad \text{(Induktionshypothese)} \\ &= 2(s+1) + 3 \end{aligned}$$

Übungsaufgabe 4.2 (WHILE-Programme)

(a) Geben Sie ein WHILE Programm P in *strikter Syntax* an, welches die Funktion $f : \mathbb{N} \dashrightarrow \mathbb{N}$ definiert durch

$$f(n) = \begin{cases} 1 & \text{falls } n < 23 \\ \perp & \text{sonst} \end{cases}$$

berechnet.

(b) Sei $M = \{n^2 \mid n \in \mathbb{N}\}$. Definiere die charakteristische Funktion $\chi_M : \mathbb{N} \to \mathbb{N}$ durch

$$\chi_M(m) = \begin{cases} 1 & \exists n \in \mathbb{N} \colon n^2 = m \\ 0 & \text{sonst.} \end{cases}$$

für alle $m \in \mathbb{N}$. Zeigen Sie, dass χ_M WHILE-berechenbar ist.

LÖSUNG: (a) Das folgende WHILE-Programm berechnet f und ist in strikter Syntax:

```
x2 = x2 + 22;
WHILE(x2 ≠ 0) {
    x2 = x2 − 1;
    x1 = x1 − 1;
}
WHILE(x1 ≠ 0) {
    x1 = x1 + 0;
}
x1 = x1 + 1;
```

Falls $x_1 < 23$, so ist der Wert von x_1 am Ende der ersten WHILE-Schleife gleich 0. In dem Fall würde die zweite WHILE-Schleife nicht ausgeführt werden, da die Anfangsbedingung $x_1 \neq 0$ nicht wahr ist. Also würde $x_1 = 1$ ausgegeben werden. Falls $x_1 \geq 23$, so ist der Wert von x_1 am Ende der ersten WHILE-Schleife ungleich 0. Somit würde das Programm in die zweite WHILE-Schleife gehen und endlos laufen.

(b) Das folgende WHILE-Programm berechnet $\chi_{M'}$:

```
x_2 = 1
x_4 = 0
WHILE(x_2 ≤ x_1) {
    x_3 = x_2 · x_2
    IF(x_3 = x_1) {x_4 = 1}
    ELSE{x_4 = x_4}
    x_2 = x_2 + 1
}
x_1 = x_4
```

In x_3 werden Quadratzahlen erzeugt. Anschließend wird geprüft, ob die erzeugte Quadratzahl gleich der Eingabe x_1 ist. Falls ja, wird x_4 auf 1 gesetzt, falls nein, ändert sich der Wert nicht. x_2 wird anschließend um 1 erhöht. Solange x_2 nicht größer als die Eingabe x_1 ist, wird die Schleife wieder ausgeführt und die nächst höhere Quadratzahl erzeugt und auf Gleichheit mit x_1 geprüft. Ist x_2 größer als die Eingabe, wird der Wert von x_4 ausgegeben.

Übungsaufgabe 4.3 (Berechenbarkeit)

Wir definieren die folgenden Mengen von Funktionen:

- $\mathbb{L} = \{f : \mathbb{N} \dashrightarrow \mathbb{N} \mid f \text{ ist LOOP-berechenbare Funktion}\}$
- $\mathbb{T} = \{f : \mathbb{N} \dashrightarrow \mathbb{N} \mid f \text{ ist total und WHILE-berechenbare Funktion}\}$
- $\mathbb{W} = \{f : \mathbb{N} \dashrightarrow \mathbb{N} \mid f \text{ ist WHILE-berechenbare Funktion}\}$
- $\mathbb{F} = \{f : \mathbb{N} \dashrightarrow \mathbb{N} \mid f \text{ ist eine Funktion}\}$

Beweisen Sie:

$$\emptyset \subsetneq \mathbb{L} \subsetneq \mathbb{T} \subsetneq \mathbb{W} \subsetneq \mathbb{F}$$

LÖSUNG: Die Inklusionen sind klar (siehe Vorlesung). Wir zeigen, dass die Inklusionen strikt sind.

- $\emptyset \subsetneq \mathbb{L}$: Die konstante Funktion $\underline{1} : \mathbb{N} \to \mathbb{N} : n \mapsto 1$ kann durch ein LOOP-Programm berechnet werden, also gilt $\mathbb{L} \neq \emptyset$.
- $\mathbb{L} \subsetneq \mathbb{T}$: Die Ackermann-Funktion ist total und WHILE-berechenbar, jedoch nicht LOOP-berechenbar.
- $\mathbb{T} \subsetneq \mathbb{W}$: Die überall undefinierte Funktion $\underline{\perp} : \mathbb{N} \dashrightarrow \mathbb{N} : n \mapsto \perp$ ist nicht total, aber WHILE-berechenbar.
- $\mathbb{W} \subsetneq \mathbb{F}$: die Menge $\mathbb{F}$ ist nicht abzählbar (vgl Ü1.1), die Menge $\mathbb{W}$ ist abzählbar (vgl. Vorlesung 1: Abzählbarkeit aller Grammatiken. Idee: Kodierung der WHILE Programme als Wort über $\mathbb{N}$; die Menge $\mathbb{N}^*$ ist abzählbar).

Hausaufgabe 4.4 (Ackermann & Co)

Wir definieren eine Funktion $h : \mathbb{N}^3 \to \mathbb{N}$ wie folgt:

$$h(n,a,b) = \begin{cases} b+1 & \text{falls } n = 0 \\ a & \text{falls } n = 1 \text{ und } b = 0 \\ 0 & \text{falls } n = 2 \text{ und } b = 0 \\ 1 & \text{falls } n \geq 3 \text{ und } b = 0 \\ h(n-1, a, h(n,a,b-1)) & \text{sonst} \end{cases}$$

für alle $n, a, b \in \mathbb{N}$. (Hinweis: h ist als *Hyper-Operator* bekannt.)

(a) Zeigen Sie, dass $h(1,a,b) = a + b$, für alle $a, b \in \mathbb{N}$. (4)

(b) Zeigen Sie, dass $h(2,a,b) = a \cdot b$, für alle $a, b \in \mathbb{N}$. (4)

LÖSUNG: (a) Beweis per Induktion über b.

Induktionsbeginn: Sei $b = 0$. Dann gilt $h(1,a,b) = h(1,a,0) = a = a + 0$.

Induktionsschritt: $h(1,a,b+1) = h(0,a,h(1,a,b)) \overset{IH}{=} h(0,a,a+b) = a + b + 1$.

(b) Beweis per Induktion über b.

Induktionsbeginn: Sei $b = 0$. Dann gilt $h(2,a,b) = h(2,a,0) = 0 = a \cdot 0$.

Induktionsschritt: $h(2,a,b+1) = h(1,a,h(2,a,b)) \overset{IH}{=} h(1,a,a \cdot b) \overset{4.4(a)}{=} a + (a \cdot b) = a \cdot (b+1)$.

Hausaufgabe 4.5 (WHILE-Programme)

(a) Geben Sie ein WHILE Programm P in *strikter Syntax* an, welches die Funktion $f : \mathbb{N}^2 \dashrightarrow \mathbb{N}$ definiert durch

$$f(m,n) = \begin{cases} 1 & \text{falls } m > n \\ \perp & \text{sonst} \end{cases}$$

berechnet. (4)

(b) Sei $T = \{3^n \mid n \in \mathbb{N}\}$. Zeigen Sie, dass die charakteristische Funktion $\chi_T : \mathbb{N} \to \mathbb{N}$ von T WHILE-berechenbar ist. (4)

LÖSUNG: (a) Das folgende strikte WHILE-Programm berechnet f.

$x_2 = x_2 + 1$
WHILE$(x_1 \neq 0)$ {
  $x_2 = x_2 - 1$
  $x_1 = x_1 - 1$
}
WHILE$(x_2 \neq 0)$ {
  $x_2 = x_2 + 0$
}
$x_1 = x_1 + 1$

Am Ende der ersten Schleife ist der Wert von x_2 gleich 0 gdw. Eingabewert von x_2 strikt kleiner war als der von x_1.

●9 ●10 ●11 auf korrektes Programm ●12 strikte Syntax

(b) Das folgende WHILE-Programm berechnet χ_T:

$x_2 = 1$
$x_3 = 0$
WHILE$(x_2 \leq x_1)$ {
  IF$(x_2 = x_1)$ $\{x_3 = 1\}$
  ELSE$\{x_3 = x_3\}$
  $x_2 = x_2 \cdot 3$
}
$x_1 = x_3$

In x_2 werden Potenzen von 3 erzeugt. Solange x_2 nicht größer als x_1 ist, wird in der Schleife geprüft, ob $x_2 = x_1$. Falls ja, wird x_3 auf 1 gesetzt, falls nein auf 0.

●13 ●14 ●15 ●16 auf korrektes Programm

Hausaufgabe 4.6 (Berechenbarkeit)

(8)

Beweisen oder widerlegen Sie die folgenden Aussagen:

(a) Die Funktion f aus Hausaufgabe 4.5 (a) ist LOOP-berechenbar.

(b) Die Funktion f aus Hausaufgabe 4.5 (a) ist WHILE-berechenbar.

(c) Die Funktion χ_T aus Hausaufgabe 4.5 (b) ist LOOP-berechenbar.

(d) Die Funktion h aus Hausaufgabe 4.4 ist LOOP-berechenbar.

LÖSUNG: (a) Die Funktion f aus Hausaufgabe 4.5 (a) ist LOOP-berechenbar. Dies ist falsch, denn mit LOOP-Programmen lassen sich nur totale Funktionen berechnen●17, f ist aber nicht total. ●18

(b) Die Funktion f aus Hausaufgabe 4.5 (a) ist WHILE-berechenbar. Offensichtlich wahr, denn wir haben ein WHILE-Programm angegeben, welches f berechnet. ●19

(c) Die Funktion χ_T aus Hausaufgabe 4.5 (b) ist LOOP-berechenbar. Die Behauptung ist wahr[20]. Als Beweis liefern wir folgendes LOOP-Progamm, welches χ_T berechnet. Wir können einfach das WHILE Programm aus Aufgabe 4.5(b) nehmen; der einzige kritische Punkt ist die WHILE-Schleife mit Bedingung $x_2 \leq x_1$. Dies lässt sich allerdings sehr einfach durch ein LOOP Programm programmieren. [21] [22]

$$\boxed{\begin{array}{l}
x_2 = 1 \\
x_3 = 0 \\
\text{LOOP}(x_1)\ \{ \\
\quad \text{IF}(x_2 = x_1)\ \{x_3 = 1\} \\
\quad \text{ELSE}\{x_3 = x_3\} \\
\quad x_2 = x_2 \cdot 3 \\
\} \\
x_1 = x_3
\end{array}}$$

(d) Die Funktion h aus Hausaufgabe 4.4 ist LOOP-berechenbar. Die Behauptung ist falsch. Die Ackermann-Funktion kann mittels dieser sogenannten Hyperoperation ausgedrückt werden (das haben wir nicht bewiesen, kann aber im Internet recherchiert werden).[23] Die Ackermann-Funktion ist jedoch nicht LOOP-berechenbar.[24]