3. Hausaufgabe im Modul „Berechenbarkeit & Komplexität“

Gruppe HA-EH-Fr-10-12-MA544-3

Aufgabe a: **LOOP- und WHILE-Programmierung**

Zuerst schauen wir uns den Basisfall $f(0, y)$ an:
Falls $y = 0$:
Da $0 = 0 \cdot k$ für alle $k \in \mathbb{N}$ ist, gilt $f(0, 0) = 1$
Falls $y > 0$:
Dann gibt es kein $k \in \mathbb{N}$, wobei $y = 0 \cdot k$. Also $f(0, y) = 0$

Als nächstes schauen wir uns die rekursive Definition an:

$$f(x + 1, y) = f(x, y) + teilbar(y, x + 1)$$

$f(x+1, y)$ wird berechnet, indem wir zu dem vorherigen Wert $f(x, y)$, 1 oder 0 addieren, abhängig davon, ob $x + 1$ ein Teiler von y ist.
Durch das rekurisve Aufrufen werden alle Teiler von y gezählt, im Bereich von 0 bis x.

Aus der Definition ergeben sich folgende besondere Fälle:
$f(x, x)$ ist die vollständige Anzahl der Teiler von x, mit Ausnahme von $x = 0$, da $f(x, 0) = x + 1$. Denn jede Zahl $x \in \mathbb{N}$ wird als Teiler von 0 gezählt, inklusive 0 (Siehe Basisfall $y = 0$).

LOOP Pseudocode mit Eingabe x_1, x_2:

```
IF x2 = 0 THEN
    x0 := x1 + 1; // Siehe Fall f(x,0)
    x1 := 0;  // Loopaufhalter
END;
x3 := 1;  // 0 wird ignoriert, siehe Basisfall y > 0
LOOP x1 DO
    x4 := x2 mod x3; // Wenn a mod b = 0, teilt b a
    IF x4 = 0 THEN
        x0 := x0 + 1; // Ergebnis wird inkrementiert
    END;
    x3 := x3 + 1;
END;
```

Aufgabe b:

Wir erkennen, dass $g(x, y)$ 1 zurückgibt, falls x weniger als y Teiler hat (Siehe $f(x, x)$ in Aufgabe a)). Es wird also 0 zurückgegeben, wenn x mindestens y Teiler besitzt.

Der μ Operator von g, definiert die Funktion $f' : \mathbb{N} \to \mathbb{N}$:

$$f'(x) = min\{n \mid g(n, x) = 0\}$$

Somit gibt $f'(x)$ die kleinste Zahl n zurück, die mindestens x Teiler hat.

Besondere Fälle:
$f'(0) = 0$, da $f(0, 0) = 1 \geq 0$
$f'(1) = 0$, da $f(0, 0) = 1 \geq 1$

Aufgabe c:

Wir fangen die Suche nach der kleinsten Zahl bei x_1 an, da keine Zahl x mehr als x Teiler hat. Also existiert keine Zahl $x < x_1$ die mindestens x_1 Teiler besitzt.

```
x2 := x1;
x3 := P_f(x2, x2);  // f(x2, x2) wird berechnet
x4 := x1 - x3;      // x4 = 0 falls f(x2, x2) >= x1

WHILE x4 != 0 DO
      x2 := x2 + 1;
      x3 := P_f(x2, x2);
      x4 := x1 - x3;
      END;
x0 := x2;
IF x1 = 1 THEN x0 := 0 END; // Siehe f'(1) in Aufgabe b
```