

Formale Sprachen und Komplexität, SS 18  
Tutoriumsblatt 8

**Aufgabe 8-1 WHILE-Programm für die partielle Wurzelfunktion**

Angenommen, das WHILE-Programm  $P$  berechnet die Funktion  $x_0 := x_1 * x_1$ .

a) Geben Sie ein WHILE-Programm an, das die partielle Wurzelfunktion

$$\begin{array}{lcl} \text{sqrt} : \mathbb{N} & \rightarrow & \mathbb{N} \\ n & \mapsto & \begin{cases} \sqrt{n} \text{ falls } n \text{ Quadratzahl} \\ \text{undef. sonst} \end{cases} \end{array}$$

berechnet.

**Lösungsvorschlag:**

Die absolute Differenz ist WHILE-berechenbar. Wir nehmen an, dass sie durch die Zuweisung  $x_0 := |a-b|$  berechnet wird.

```
q := x_1 + 0           //aus q soll die Wurzel gezogen werden
z := 1;               //Differenztester
i := 0;               //i wird das Ergebnis
WHILE z != 0 DO
    x1 := i + 0;      //Eingabe fuer P
    P;
    z := |q - x0|;    //ist x0 = i^2 = q?
    i := i + 1;      //mit naechstem i testen
END
x_0 := i - 1;
```

b) Hält Ihr Programm, falls  $n$  keine Quadratzahl ist? Falls nicht, was müssten Sie ändern, damit  $\text{sqrt}$  nicht nur partiell, sondern total definiert ist?

**Lösungsvorschlag:**

Es gibt viele Möglichkeiten das zu realisieren, eine davon: Ersetze die Zeile

```
z := |q - x0|; //ist x0 = i^2 = q?
```

durch

```
z := q - x0; //ist x0 < i^2 = q?
```

Da  $x_0$  wächst, ist  $x_0 < q$  nach endlich vielen Schleifendurchläufen erfüllt. Dieses neue Programm berechnet zu jedem  $q$  die Wurzel der nächst größeren Quadratzahl.  $\lceil \sqrt{q} \rceil$

## Aufgabe 8-2 GOTO-Programme

Schreiben Sie ein GOTO-Programm, das

a)  $x_0 := x_1 + x_2$  berechnet.

### Lösungsvorschlag:

```
M1 x0 := x1 + 0;
M2 x9 := x2 + 0;          # x9 ist kopie von x2
M3 IF x9 = 0 THEN GOTO M7; # x9 dekrementieren und x0 inkrementieren
                          # bis x9 null ist
M4   x9 := x9 - 1;
M5   x0 := x0 + 1;
M6 GOTO M3;
M7 HALT
```

b)  $x_0 := x_1 * x_2$  berechnet.

### Lösungsvorschlag:

```
Mn   x9 := x2 - 1;          # marke um x9 zu setzen (Mn wie neun) de
                          # multiplikationszaehler
Mz   x10 := x1 + 0;         # marke um x10 zu setzen (Mz wie zehn)
                          # der additionszaehler
Ma   IF x10 = 0 THEN GOTO Mm; # addierungs loop start (Ma wie addieren)
Ma1  x10 := x10 - 1;
Ma2  x0 := x0 + 1;
Ma3  GOTO Ma;              # addierungs loop ende
Mm   IF x9 = 0 THEN GOTO Mh; # multiplications loop start
Mm1  x9 := x9 - 1;
Mm2  GOTO Mz;              # multiplications loop ende
Mh   HALT
```