

GOTO-Programme

- (a) Terminieren GOTO-Programme immer? Begründe Deine Antwort.

GOTO-Programme terminieren nicht immer.

Variante 1: Die Menge der GOTO-Programme ist identisch mit der Menge der WHILE-Programme. Da WHILE-Programme partielle Funktionen beschreiben und diese nicht für alle Eingaben terminieren, terminieren GOTO-Programme ebenfalls nicht für alle Eingaben.

Variante 2: Die charakteristische Funktion einer semi-entscheidbaren Sprache ist Turing- bzw. GOTO-berechenbar, d.h. zu jeder semi-entscheidbaren Sprache gibt es eine Turing-Maschine. GOTO-Programme können Turing-Maschinen simulieren. Da hier von einer nur semi-entscheidbaren Sprache ausgegangen wird, terminiert das GOTO-Programm nicht, falls die Eingabe x kein Element der Sprache ist.

- (b) Gebe ein GOTO-Programm an, dass die Summe dreier Zahlen berechnen.

```
1  Eingabe x_1, x_2, x_3;
2  x_0 := x_1;
3  IF x_2 = 0 GOTO Z6;
4    x_0 := x_0 + 1;
5    x_2 := x_2 - 1;
6  GOTO Z2;
7  IF x_3 = 0 GOTO Z10;
8    x_0 := x_0 + 1;
9    x_3 := x_3 - 1;
10 GOTO Z6;
11 END;
12 Ausgabe: x_0
```

- (c) Gegeben ist das GOTO-Programm:

```
1  x_4 := x_1;
2  IF x_4 = 0 GOTO Z10;
3  x_5 := x_2;
4  IF x_5 = 0 GOTO Z8;
5  x_3 := x_3 + 1;
6  x_5 := x_5 - 1;
7  GOTO Z4;
8  x_4 := x_4 - 1;
9  GOTO Z2;
10 x_5 := x_5 - 1
```

- (i) Was berechnet das Programm?

$$f(n, m) = n * m$$

- (ii) Übertrage das Programm in ein WHILE-Programm.

```

1  Eingabe x_1, x_2 :
2  x_4 := x_1;
3  WHILE x_4 <> 0 DO
4      x_5 := x_2;
5      WHILE x_5 <> 0 DO
6          x_3 := x_3 + 1;
7          x_5 := x_5 - 1
8      END;
9  x_4 := x_4 - 1
10 END
11 Ausgabe x_0

1  Eingabe : x_1, x_2
2  x_0 := mult ( x_1, x_2 );
3  Ausgabe : x_0

```