

## WHILE-berechenbar [2 hoch x, ggT, if]

Bestimme jeweils, ob die angegebene Funktion WHILE-berechenbar ist:

(a)  $x \rightarrow 2^x$

```

1  erg = 1;
2  WHILE x != 0 DO
3      erg = erg * 2;
4      x = x - 1;
5  END;
6  return erg;
```

(b)  $\text{ggT}(n, m)$ ,

also der größte gemeinsame Teiler. Sie dürfen die (ganzzahligen) Operationen  $+$ ,  $-$ ,  $*$  und  $/$  verwenden, wobei das Minus, wie üblich, eingeschränkt ist.

Es bietet sich an, zunächst die modulo Operation

```

1  x_i := x_j % x_k
```

durch folgendes WHILE-Programm zu definieren:

```

1  x_n+1 := x_j / x_k;
2  x_n+2 := x_n+1 * x_k;
3  x_i := x_j - x_n+2;
```

Wobei  $x_{n+1}$  und  $x_{n+2}$  im Rest des Programmes nicht verwendet werden sollen. Mit der Modulo Operation kann man nun z. B. einfach den euklidischen Algorithmus verwenden (Eingabe seien  $x_1$  und  $x_2$ , Ausgabe ist  $x_1$ ):

```

1  WHILE x_2 != 0 DO
2      x_3 := x_1 % x_2;
3      x_1 := x_2 + 0;
4      x_2 := x_3 + 0;
5  END
```

(c)  $\text{if } x_i \neq 0 \text{ then } P_1 \text{ else } P_2 \text{ fi}$

mit der üblichen Semantik. Als Nachweis kann jeweils ein WHILE-Programm angegeben werden.

Sei  $x_n$  die höchste in  $P_1$  bzw.  $P_2$  vorkommende Variable (o. E.  $i \leq n$ ).

```

1  x_n+1 := x_i + 0;
2  x_n+2 := 1;
3  WHILE x_n+1 != 0 DO
4      x_n+1 := 0;
5      x_n+2 := 0;
6      P_1;
7  END
8  WHILE x_n+2 != 0 DO
9      x_n+2 := 0;
```

```
10   P_2;  
11   END
```