## WHILE-berechenbar

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

(a) $x \to 2^x$

```
erg = 1;
WHILE x != 0 DO
  erg = erg * 2;
  x = x - 1;
END;
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

```
x_i := x_j % x_k
```

durch folgendes WHILE-Programm zu definieren:

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

Wobei $x_{n+1}$ und $x_{n+1}$ 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 $x_1$:

```
WHILE x_2 != 0 DO
  x_3 := x_1 % x_2;
  x_1 := x_2 + 0;
  x_2 := x_3 + 0;
END
```

(c) `if x_i != 0 then P_1 else P_2 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. $i \leq n$).

```
x_n+1 := x_i + 0;
x_n+2 := 1;
WHILE x_n+1 != 0 DO
  x_n+1 := 0;
  x_n+2 := 0;
  P_1;
END
WHILE x n+2 6 = 0 DO
  x_n+2 := 0;
  P_2;
```

```
END
```