Aufgabenblatt Nr. 4 inf11 26.01.2021

10. Der Algorithmus **PRIMZAHLTEST**

Definition: Eine natürliche Zahl n heißt Primzahl genau dann, wenn sie nur durch 1 und durch sich selbst jeweils ohne Rest teilbar ist.

Aufgabe: Konzipiere einen Algorithmus (als Struktogramm und als Python-Programm), der nach Eingabe einer natürlichen Zahl n entscheidet, ob n die Primzahleigenschaft hat.

Hinweis: Teste für alle in Frage kommenden Teiler (Divisoren) t, ob n % t gleich 0 ist.

11. **BOOLESCHE VARIABLE** oder **BOOLESCHE TERME** können nur zwei Werte annehmen: **True** oder **False**.

Die Verknüpfungen **and** und **or** sowie die Operation **not** werden bekanntlich jeweils über eine Wahrheitstafel definiert.

Wir verwenden folgende abkürzende Schreibweisen (a, b, c sind Boolesche Variable):

a and b = a · b = a b
a or b = a + b
not a = ¬ a

Dabei gelte auch die aus der Algebra bekannte Vereinbarung: "Punkt vor Strich", d. h.
a + (b · c) = a + b · c = a + b c

Eine Auswahl von Rechenregeln für Boolesche Variable:

Kommutativgesetze

(1) a + b = b + a **(1′) a · b = b · a**

Assoziativgesetze

(2) a + (b + c) = (a + b) + c **(2′) a · (b · c) = (a · b) · c**

Distributivgesetze

(3) a · (b + c) = a · b + a · c **(3′) a + b · c = (a + b)·(a + c)**

Beweis von (3):

a	b	c	b + c	a(b + c)	ab	ac	ab + ac
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Da die Spalten zu **a(b + c)** und **ab + ac** übereinstimmen, gilt: **a(b + c) = ab + ac** .

Aufgaben: a) Beweise die Rechengesetze (2) und (3′).

b) Zeige: (a · b) + c ≠ a · (b + c)