

15. **BOOLESCHES TERME**

Schreibweisen für die Negation:

$$\text{not } a = \neg a = \bar{a}$$

Wir ergänzen die in Aufgabe 11 (Blatt 4, 26.01.2021) für Boolesche Variable formulierten Rechenregeln

- Kommutativgesetze (1) und (1')
- Assoziativgesetze (2) und (2')
- Distributivgesetze (3) und (3')

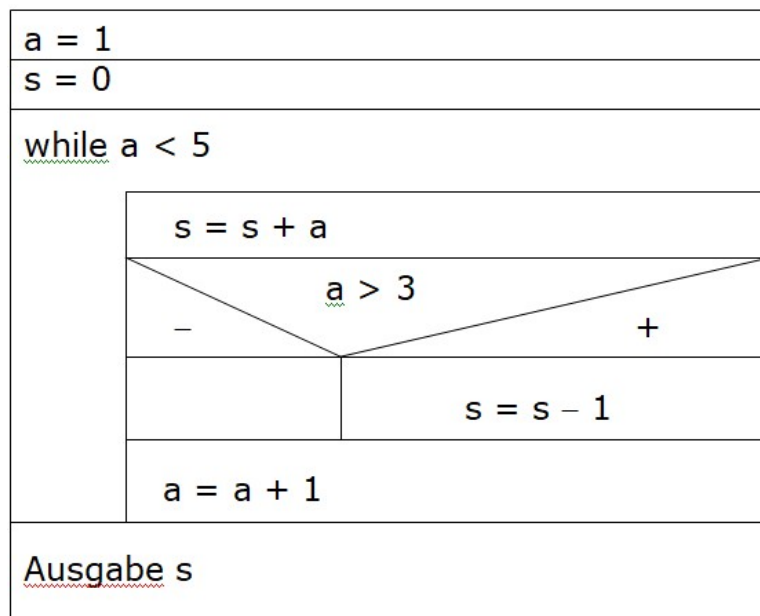
um die beiden Gesetze von de Morgan:

$$(4) \quad \overline{a \cdot b} = \bar{a} + \bar{b}$$

$$(4') \quad \overline{a + b} = \bar{a} \cdot \bar{b}$$

Aufgabe: Beweise (4) (*Hinweis: Wahrheitstafel!*)

16. Erstelle einen in Python geschriebenen Quelltext zu folgendem Struktogramm:

17. In der Zusammenfassung **MinSuche_SelectionSort.pdf** werden die Algorithmen **MinSuche**, **MinSuche2** und **SelectionSort** noch einmal erläutert.

Aufgabe: Formuliere bei

- a) MinSuche die for-Schleife zur Bestimmung des kleinsten Elements,
- b) SelectionSort die beiden for-Schleifen zum Sortieren

jeweils als while-Schleifen und überprüfe die so erhaltenen Python-Programme anhand von Testläufen.

18. Erstelle jeweils ein Struktogramm zu **MinSuche** und **SelectionSort** (Hinweis: verwende bevorzugt die Versionen mit while-Schleife, Aufgabe 17; für die Zuweisung von Zufallszahlen an die Komponenten des arrays a genügt es zu schreiben: „Zuweisung von Zufallszahlen an a[0], . . . , a[n-1]“)

19. Freiwillige Aufgabe:

Der Algorithmus SelectionSort (Quelltext: SelectionSort_04-03-2021.py) vertauscht die Inhalte der Speicherplätze a[j] und a[i], j < i < n, immer dann, wenn ein kleineres a[i] als a[j] gefunden wurde; hier gibt es noch Optimierungspotential, um die Rechenzeit insgesamt zu verkürzen. Ergreife diese Möglichkeit und teste! (Die Laufzeit zum Sortieren lässt sich für große Werte von n etwa halbieren.)