

---

## Algorithmen und Datenstrukturen (EI)

---

### Aufgabe 1

Markieren Sie, ob folgende Aussagen in voller Allgemeinheit gelten (J:ja/wahr, N:nein/falsch).

- |  |   |
|--|---|
| Betrachtet man ein Alphabet $\Sigma = \{a, b\}$ , so ist die Sprache $L = \Sigma^*$ regulär        | <input type="checkbox"/> J <input type="checkbox"/> N |
| Seien $L_1$ und $L_2$ regulär, dann ist $L_1 \cap L_2$ auch regulär                                | <input type="checkbox"/> J <input type="checkbox"/> N |
| Sei $L$ kontextfrei, dann ist $\bar{L}$ auch kontextfrei   | <input type="checkbox"/> J <input type="checkbox"/> N |
| Arithmetische Ausdrücke können durch eine Chomsky-3-Grammatik erzeugt werden                       | <input type="checkbox"/> J <input type="checkbox"/> N |
| QuickSort ist ein optimales vergleichsbasiertes Sortierverfahren                                   | <input type="checkbox"/> J <input type="checkbox"/> N |
| HeapSort hat quadratische Laufzeit im worst-case   | <input type="checkbox"/> J <input type="checkbox"/> N |
| Ein AVL-Baum mit $n$ Knoten hat Tiefe $\Omega(n)$  | <input type="checkbox"/> J <input type="checkbox"/> N |
| Mit Hilfe der Chomsky-Normalform kann das Wortproblem für kontextfreie Sprachen entschieden werden | <input type="checkbox"/> J <input type="checkbox"/> N |

### Aufgabe 2

Gegeben sei die Grammatik

$$\begin{aligned} G := & (\{S, T, A, B\}, \\ & \{a, b\}, \\ & \{S \rightarrow AS, S \rightarrow T, A \rightarrow aB, aBT \rightarrow Tab, aBT \rightarrow ab\}, \\ & S) \end{aligned}$$

1. Geben Sie eine Grammatik vom Typ 3 an, die  $L(G)$  erzeugt.
2. Geben Sie einen DFA an, der  $L(G)$  erkennt.

### Aufgabe 3

Sei  $G = (V, \Sigma, P, S)$  eine kontextfreie Grammatik in Chomsky-Normalform ohne nutzlose Variablen und mit  $n$  Produktionen. Zeigen Sie, dass die Anzahl der Produktionen bei der Überführung in Greibach-Normalform in Abhängigkeit von  $n$  exponentiell wachsen kann.

## Aufgabe 4

Entwerfen Sie einen Algorithmus, der in einer Liste aus  $n$  Elementen das zweitgrösste Element findet. Der Algorithmus soll mit weniger als  $2n - 3$  Vergleichen auskommen.