

---

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
--------------------	----------------	----------------------

---

Kennzahl: \_\_\_\_\_

Kennwort: \_\_\_\_\_

Arbeitsplatz-Nr.: \_\_\_\_\_

---

**Herbst  
2021**

**46115**

**Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen**  
**— Prüfungsaufgaben —**

---

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik/Algorithmen/Datenstrukturen**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **14**

---

Bitte wenden!

Thema Nr. 1  
(Aufabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

**Teilaufgabe I: Algorithmen**

**Aufgabe 1 (Algorithmenanalyse)**

[15 PUNKTE]

Gegeben sei ein Feld A von ganzen Zahlen. Das Feld habe n Elemente A[1] bis A[n]. Der folgende Algorithmus sei gegeben:

```
1 function mystery : boolean
2   var critical : boolean; y, i, j : integer;
3   begin
4     critical := false;
5     for i:=1 to n-1 do
6       begin
7         y := A[i];
8         for j:=n downto i+1 do
9           begin
10            if A[j]==y then critical := true;
11          end
12        end
13      return critical;
14    end
```

- (a) Wenden Sie den Algorithmus für das folgende Feld an. Es genügt anzugeben, welchen Rückgabewert der Algorithmus berechnet.

Index	1	2	3	4
Wert	5	7	6	9

- (b) Wenden Sie den Algorithmus für das folgende Feld an. Es genügt anzugeben, welchen Rückgabewert der Algorithmus berechnet.

Index	1	2	3	4
Wert	5	5	6	7

Fortsetzung nächste Seite!

(c) Im Folgenden soll die Laufzeit des Algorithmus in Bezug auf die Eingabegröße  $n$  analysiert werden. Gezählt werden sollen *ausschließlich Zuweisungen an die Variable critical in Zeile 10* des Algorithmus.

- i) Wie viele Zuweisungen an die Variable critical in Zeile 10 erfolgen im *worst case*?  
Wie muss das Feld gestaltet sein, damit der *worst case* eintritt?
- ii) Wie viele Zuweisungen an die Variable critical in Zeile 10 erfolgen im *best case*?  
Wie muss das Feld gestaltet sein, damit der *best case* eintritt?

## Aufgabe 2 (Sortiervverfahren)

[30 PUNKTE]

In der folgenden Aufgabe soll ein Feld A von ganzen Zahlen aufsteigend sortiert werden. Das Feld habe  $n$  Elemente  $A[1]$  bis  $A[n]$ . Der folgende Algorithmus sei gegeben:

```
1 var A : array[1..n] of integer;
2
3 procedure bubblysort
4 var i, j, tmp : integer;
5     magic : boolean;
6 begin
7     magic := true;
8     j := n-1;
9     while (j >= 1 and magic) do
10    begin
11        magic := true;
12        for i := 1 to j do
13            begin
14                if A[i] > A[i+1] then
15                    begin
16                        tmp := A[i];
17                        A[i] := A[i+1];
18                        A[i+1] := tmp;
19                        magic := true;
20                    end
21            end
22        j := j-1;
23    end
24 end
```

Fortsetzung nächste Seite!

- a) Sortieren Sie das folgende Feld A mittels des Algorithmus. Geben Sie die Belegung des Feldes nach jedem Durchlauf der inneren Schleife in einer neuen Zeile auf Ihrem Bearbeitungsbogen an.

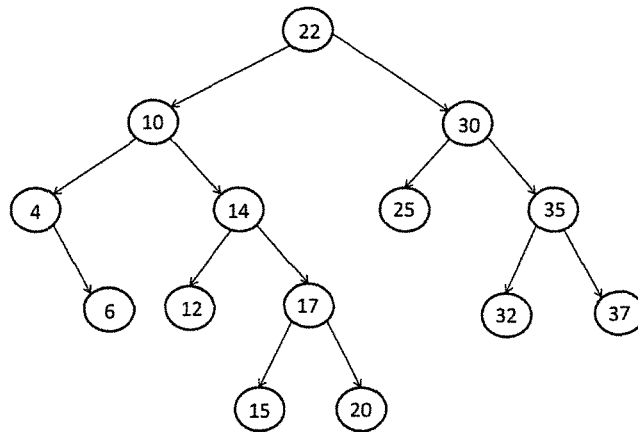
Index	1	2	3	4	5
Wert	9	5	7	3	1

Die folgenden Zeilenangaben beziehen sich auf den gegebenen Algorithmus.

- b) Geben Sie die *genaue* Anzahl der in Zeile 14 durchgeführten *Vergleichsoperationen* zwischen Feldelementen als Funktion  $f$  der Eingabegröße  $n$  an. Begründen Sie Ihre Antwort. (Beachten Sie, dass hier die *genaue* Anzahl gefragt ist und *nicht* die asymptotische Komplexität.)
- c) Wie muss das Feld beschaffen sein, damit möglichst wenige Vertauschungsoperationen (Zeilen 16–18) durchgeführt werden? Begründen Sie Ihre Antwort.
- d) Welche Auswirkung auf das Ergebnis des Algorithmus hätte es, wenn die Bedingung in Zeile 14 wie folgt lauten würde? Begründen Sie Ihre Antwort.  
`if A[i] < A[i+1] then`
- e) Welche Auswirkung auf das Verhalten des Algorithmus hätte es, wenn die Bedingung in Zeile 14 wie folgt lauten würde? Erläutern Sie Ihre Antwort.  
`if A[i] >= A[i+1] then`
- f) Welche Auswirkung auf das Ergebnis des Algorithmus hätte es, wenn die Bedingung der `while`-Schleife in Zeile 9 wie folgt lauten würde? Begründen Sie Ihre Antwort.  
`while (j>1 and magic) do`
- g) Welche Auswirkung auf das Ergebnis des Algorithmus hätte es, wenn die Variable `magic` in Zeile 7 auf den Wert `false` initialisiert werden würde? Begründen Sie Ihre Antwort.
- h) Welche Auswirkung auf das Ergebnis des Algorithmus hätte es, wenn die Zuweisung in Zeile 19 des Algorithmus `magic := false` lauten würde? Begründen Sie Ihre Antwort.
- i) Welche Auswirkung auf das Ergebnis des Algorithmus hätte es, wenn die Zuweisung in Zeile 11 des Algorithmus `magic := false` lauten würde? Begründen Sie Ihre Antwort.

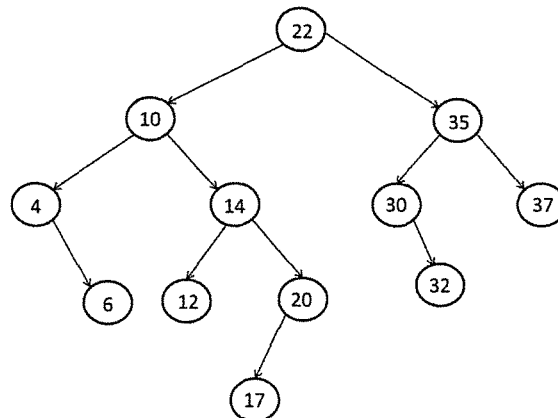
**Aufgabe 3 (AVL-Bäume)****[18 PUNKTE]**

a) Es sei der folgende AVL-Baum mit Schlüsseln aus  $N$  gegeben:



Fügen Sie den Schlüssel **16** in den AVL-Baum ein. Rebalancieren Sie anschließend den Baum so, dass die AVL-Eigenschaft wieder erreicht wird. Zeichnen Sie den Baum nach jeder Einfach- und Doppelrotation und benennen Sie die Art der Rotation (Links-, Rechts-, Links-Rechts-, oder Rechts-Links-Rotation).

b) Es sei der folgende AVL-Baum mit Schlüsseln aus  $N$  gegeben:

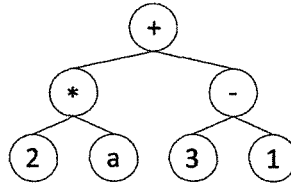


Löschen Sie den Schlüssel **37** aus diesem AVL-Baum. Rebalancieren Sie anschließend den Baum so, dass die AVL-Eigenschaft wieder erreicht wird. Zeichnen Sie den Baum nach jeder Einfach- und Doppelrotation und benennen Sie die Art der Rotation (Links-, Rechts-, Links-Rechts-, oder Rechts-Links-Rotation). Argumentieren Sie jeweils über die Höhenbalancen der Teilbäume.

Fortsetzung nächste Seite!

**Aufgabe 4 (Baumtraversierung)****[17 PUNKTE]**

- a) Gegeben sei folgender geordneter Binärbaum mit der Wurzel „+“ und den jeweils linken und rechten Nachfolgerknoten.



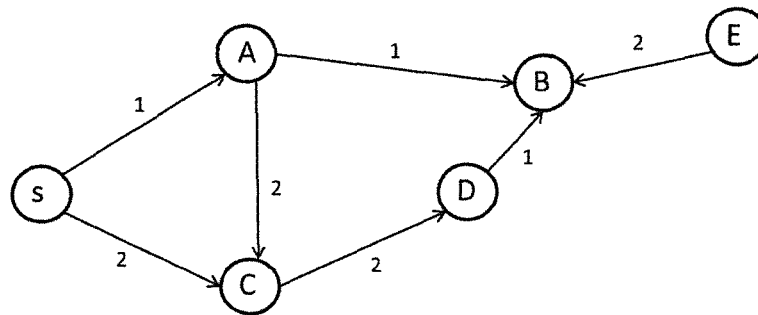
Traversieren Sie den Baum und geben Sie die besuchten Knoten in der jeweils geforderten Reihenfolge an. Falls es mehrere Nachfolger gibt, besuchen Sie zuerst den linken und dann den rechten Nachfolger.

- i) Geben Sie die besuchten Knoten in der Reihenfolge *preorder* an.
  - ii) Geben Sie die besuchten Knoten in der Reihenfolge *inorder* an.
  - iii) Geben Sie die besuchten Knoten in der Reihenfolge *postorder* an.
- b) Zeigen oder widerlegen Sie: Es gibt nichtleere geordnete Binärbäume, bei denen die Traversierung *inorder* und *postoder* das gleiche Ergebnis liefert.

Fortsetzung nächste Seite!

**Aufgabe 5 (Kürzeste Wege)****[10 PUNKTE]**

Gegeben ist der folgende gerichtete Graph mit Kanten- bzw. Entfernungsgewichten.



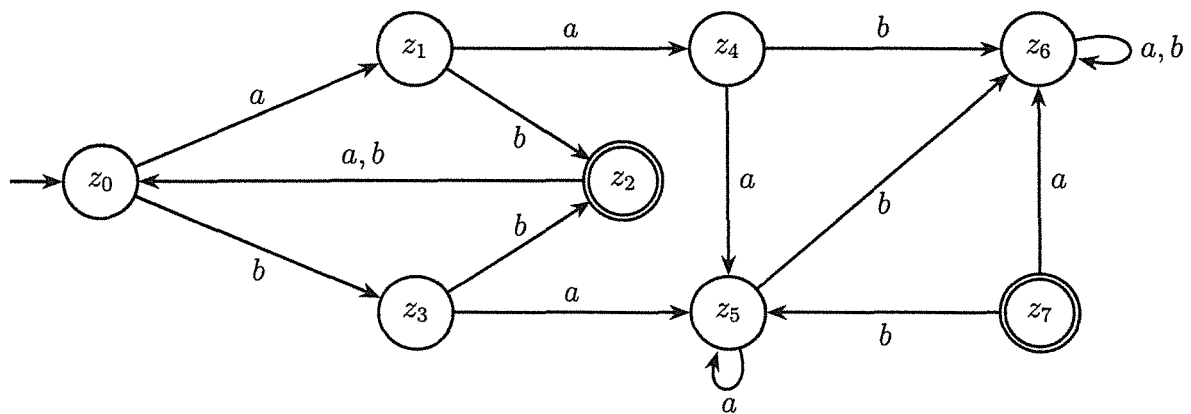
Bestimmen Sie die kürzesten Wege des Graphen vom Startknoten  $s$  aus zu allen übrigen Knoten mit Hilfe des Algorithmus von Dijkstra. Erstellen Sie dazu auf Ihrem Bearbeitungsbogen eine Tabelle der folgenden Art und markieren Sie in jeder Zeile den jeweils als nächstes zu betrachtenden Knoten. Tragen Sie das Endergebnis für alle Knoten in der untersten Zeile ein.

	s	A	B	C	D	E
Initialisierung	0	$\infty$	$\infty$	$\infty$	$\infty$	$\infty$
1. Iteration						
Endergebnis						

Fortsetzung nächste Seite!

Teilaufgabe II: Theoretische Informatik**Aufgabe 1 (Reguläre Sprachen und endliche Automaten)****[40 PUNKTE]**

- a) In einen Geldwechsellautomat für Euromünzen können 1- und 2-Euro-Münzen eingeworfen werden. Der Automat akzeptiert eine Folge solcher Münzen, wenn diese in Euro-Scheine wechselbar ist, was genau dann der Fall ist, wenn der Betrag durch 5 teilbar ist. Die Steuerung des Geldwechsellautomaten verwendet einen deterministischen endlichen Automaten, der die Eingabe als Folge von 1- und 2-Euro-Münzen (d. h. als Wort über dem Alphabet  $\{1,2\}$ ) erhält, und genau die nicht-leeren Folgen akzeptiert, deren jeweilige Summe durch 5 teilbar ist.
- Geben Sie einen **deterministischen endlichen Automaten** als **Zustandsgraph** an, der für die interne Steuerung des Geldwechsellautomaten verwendet werden kann (d. h. der Automat akzeptiert genau alle nicht-leeren Folgen von 1- und 2-Euro-Münzen, deren jeweilige Summe durch 5 teilbar ist).
  - Geben Sie einen **regulären Ausdruck** an, der die akzeptierte Sprache des Geldwechsellautomaten erzeugt.
- b) Berechnen Sie für den folgenden deterministischen endlichen Automaten, der Worte über dem Alphabet  $\Sigma = \{a, b\}$  verarbeitet, den Minimalautomaten, d. h. einen deterministischen endlichen Automaten, der die gleiche Sprache akzeptiert und eine minimale Anzahl an Zuständen benutzt. Erläutern Sie Ihre Berechnung, indem Sie z. B. eine Minimierungstabelle angeben, und geben Sie den Minimalautomaten als Zustandsgraph an.



Fortsetzung nächste Seite!



**Aufgabe 2 (Chomsky-Hierarchie, Formale Grammatiken)****[30 PUNKTE]**

a) Die Sprache  $L_1$  sei definiert als

$$L_1 = \{a^i cb^j c \mid 0 \leq i < j\}$$

Beweisen Sie, dass die Sprache  $L_1$  nicht regulär ist.

b) Die kontextfreie Grammatik  $G = (V, \Sigma, P, S)$  sei definiert durch:

$$V = \{S, A, B, C, D, E, F, G, H\}$$

$$\Sigma = \{a, b\}$$

$$P = \{A \rightarrow a \mid AA,$$

$$B \rightarrow b \mid BB,$$

$$C \rightarrow AD \mid AB,$$

$$D \rightarrow CB,$$

$$E \rightarrow AE \mid BF,$$

$$F \rightarrow BF \mid b,$$

$$G \rightarrow AA,$$

$$H \rightarrow BB,$$

$$S \rightarrow GH \mid AD \mid AE \}$$

$S$  ist das Startsymbol

i) Entscheiden Sie, ob

$$aaabbb \in L(G)$$

gilt, indem Sie den Algorithmus von Cocke, Younger und Kasami (CYK-Algorithmus) ausführen. Geben Sie die dabei entstehende Tabelle und die Ausgabe des Algorithmus an und dokumentieren Sie den Ablauf des Algorithmus.

ii) Welche nicht-leeren, echten Präfixe von  $aaabbb$  (d. h. welche der Worte  $a$ ,  $aa$ ,  $aaa$ ,  $aaab$ ,  $aaabb$ ) liegen in  $L(G)$ ? Begründen Sie Ihre Antwort mithilfe der im vorherigen Aufgabenteil erstellten Tabelle.

Fortsetzung nächste Seite!

**Aufgabe 3 (Berechenbarkeit)****[20 PUNKTE]**

Beweisen oder widerlegen Sie jeweils für die folgenden Sprachen, dass diese **entscheidbar** sind. Die Gödel-Nummer der Turingmaschine  $M$  wird dabei mit  $\langle M \rangle$  bezeichnet.

- a)  $L_2 = \{\langle M \rangle \mid M \text{ hält für irgendeine Eingabe } w \in \{a, b\}^* \text{ nach nicht mehr als 100 Schritten}\}$
- b)  $L_3 = \{\langle M \rangle \mid M \text{ macht bei Eingabe } aba \text{ mindestens 200 Schritte und } M \text{ hält bei Eingabe } aba\}$

Thema Nr. 2  
(Aufabengruppe)

Es sind alle Aufgaben dieser Aufabengruppe zu bearbeiten!

**Teilaufgabe I: Algorithmen**

**Aufgabe 1 (  $\mathcal{O}$  -Notation)**

[20 PUNKTE]

- a) Ordnen Sie die folgenden Funktionen gemäß ihrem asymptotischen Wachstum. Geben Sie auch an, wenn zwei Funktionen asymptotisch gleich schnell wachsen.

$$2^{4 \log n}, \quad \sqrt{n}, \quad 4^{2 \log n}, \quad n^5, \quad 1000n, \quad 2^{\sqrt{n}}, \quad n^2 \log n, \quad n^2$$

- b) Gilt zwischen den folgenden Mengen eine Teilmengenbeziehung? Ist diese echt? Geben Sie einen formalen Beweis an.

$$\mathcal{O}(n^2 \log(n)) \quad \mathcal{O}(\sqrt{n^5})$$

**Aufgabe 2 (Algorithmen-Entwurf)**

[27 PUNKTE]

Betrachten Sie folgenden Algorithmus in Java-Code:

```
public static boolean f(int[] a) {
    int x = 1;
    for (int i = 0; i < a.length - 1; ++i) {
        x = x * a[i];
        int y = 1;
        for (int j = i + 1; j < a.length; ++j) {
            y = y * a[j];
        }
        if (x == y) {
            return true;
        }
    }
    return false;
}
```

- a) Geben Sie zwei konkrete Arrays  $a$  und  $b$  je der Länge 6 an mit  $f(a) = \text{true}$  und  $f(b) = \text{false}$ .
- b) Was berechnet  $f$  allgemein bei der Eingabe eines Arrays  $a$  aus  $n$  ganzen Zahlen?
- c) Was ist die genaue Anzahl von Zuweisungen an  $y$ , die  $f$  durchführt, in Abhängigkeit von  $n = a.length$  auf einem Eingabearray  $a$  im Best Case und im Worst Case? Welche asymptotische Laufzeit von  $f$  ergibt sich daraus?
- d) Geben Sie eine Funktion  $g$  in Pseudocode oder einer objektorientierten Programmiersprache Ihrer Wahl an, welche bei gleicher Eingabe wie in  $f$  immer die gleiche Ausgabe wie  $f$  liefert, aber eine optimale asymptotische worst-case Laufzeit hat. Geben Sie diese Laufzeit an und begründen Sie kurz die Korrektheit Ihrer Funktion.

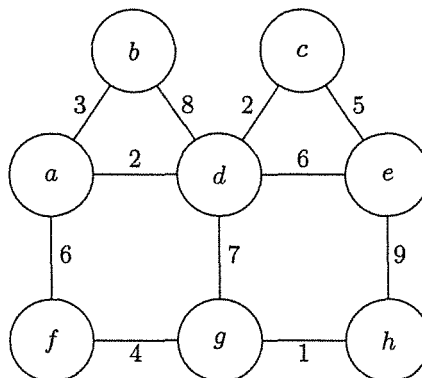
Fortsetzung nächste Seite!

**Aufgabe 3 (Min-Heaps)****[18 PUNKTE]**

Gegeben sei folgender Min-Heap H:

$$[-5, 6, 3, 11, 18, 10, 8, 13, 17, 19]$$

- Fügen Sie in den Heap H den Wert 15 ein. Beschreiben Sie alle nötigen Schritte, um die Heap-Struktur wiederherzustellen. Zeichnen Sie den Heap nach der Einfügeoperation.
- Führen Sie auf dem Heap H `deletemin` durch. Beschreiben Sie alle nötigen Schritte, um die Heap-Struktur wiederherzustellen. Zeichnen Sie den Heap nach der Löschoption. Gehen Sie dabei vom ursprünglichen Suchbaum aus und *nicht* von ihrem Ergebnis aus Teilaufgabe a).
- Geben Sie jeweils an, ob die folgenden Aussagen wahr oder falsch sind und begründen Sie Ihre Antworten kurz.
  - In einem Min-Heap hat der Knoten mit dem größten Element immer maximale Tiefe.
  - Das drittkleinste Element in einem Min-Heap ist nicht notwendigerweise ein Kind der Wurzel.

**Aufgabe 4 (Minimale Spannbäume)****[25 PUNKTE]**Gegeben sei folgender ungerichteter Graph  $G = (V, E)$  mit Kantengewichten  $\omega: E \rightarrow \mathbb{N}$ .

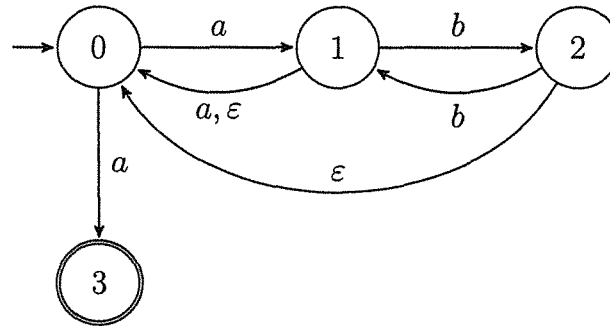
- Berechnen Sie mithilfe des Jarník-Prim-Algorithmus einen minimalen Spannbaum  $T$  von  $G$  beginnend von Knoten  $a$ . Bearbeiten Sie Knoten bei Wahlfreiheit in alphabetischer Reihenfolge.  
Erstellen Sie dazu eine Tabelle mit zwei Spalten und stellen Sie jeden einzelnen Schritt des Verfahrens in einer eigenen Zeile dar. Geben Sie in der ersten Spalte denjenigen Knoten  $v$  an, der vom Algorithmus als nächstes in  $T$  aufgenommen wird (der neue „schwarze“ Knoten). Führen Sie in der zweiten Spalte alle anderen vom aktuellen Baum  $T$  direkt erreichbaren Knoten  $v$  (die „grauen“ Knoten) auf, die noch nicht im aktuellen Baum enthalten sind. Notieren Sie Knoten immer zusammen mit deren Vorgänger und der Distanz zum Vorgänger als Tripel wie folgt: (Knoten, Vorgänger, Distanz).
- Zeichnen Sie den minimalen Spannbaum aus Teilaufgabe a) und geben Sie sein Gewicht an.
- Sei  $G'$  der Graph, den man aus  $G$  erhält, indem man die Kante  $\{d, f\}$  mit Gewicht 0 hinzufügt. Was ist das Gewicht eines minimalen Spannbaums von  $G'$ ? Begründen Sie Ihre Antwort.

Fortsetzung nächste Seite!

**Teilaufgabe II: Theoretische Informatik****Aufgabe 1 (Reguläre Sprachen)**

[30 PUNKTE]

Betrachten Sie den nichtdeterministischen Automaten mit  $\varepsilon$ -Übergängen  $\mathcal{A}$ , der durch den folgenden Zustandsgraphen veranschaulicht ist:



- Ist  $L(aa^*) \subseteq L(\mathcal{A})$ ? Begründen Sie Ihre Antwort.
- Ist  $L(aa^*(bbb)^*aa^*) \subseteq L(\mathcal{A})$ ? Begründen Sie Ihre Antwort.
- Wandeln Sie  $\mathcal{A}$  in einen deterministischen endlichen Automaten um. Verwenden Sie dafür die Potenzmengenkonstruktion.
- Ist  $\{w_1 x w_2 \mid w_1, w_2 \in \{a, b\}^*, x \in \{a, b\}, |w_1| = |w_2|\}$  eine reguläre Sprache? Beweisen Sie Ihre Antwort.
- Ist  $\{w_1 x w_2 \mid w_1, w_2 \in \{a\}^*, x \in \{a, b\}, |w_1| = |w_2|\}$  eine reguläre Sprache? Beweisen Sie Ihre Antwort.

**Aufgabe 2 (Kontextfreie Sprachen)**

[30 PUNKTE]

- Zeigen Sie mit dem CYK-Algorithmus, dass das Wort  $w = abbabbab$  in der Sprache der Grammatik  $G = (\{B, C, S\}, \{a, b\}, P, S)$  mit

$$\begin{aligned}
 P = \{ & S \rightarrow CB \mid b, \\
 & B \rightarrow SS \mid CS, \\
 & C \rightarrow CC \mid BC \mid a \}
 \end{aligned}$$

enthalten ist.

Geben Sie zwei verschiedene Ableitungsbäume von  $w$  für die Grammatik  $G$  an.

- Ist

$$L_{eq} = \{w w c w^R w^R \mid w \in \{a, b\}^*\}$$

eine kontextfreie Sprache? Beweisen Sie Ihre Antwort.

*Hinweis:* Die Umkehrung  $w^R$  eines Wortes  $w$  ist das Wort, das man durch das Lesen von  $w$  von rechts nach links erhält. So sind  $(abab)^R = baba$  und  $(abcd)^R = dcba$ .

- Ist  $L = \{a^n b^m c^m d^n \mid n, m \in \{0, 1, 2, \dots\}\}$  eine kontextfreie Sprache? Beweisen Sie Ihre Antwort.

Fortsetzung nächste Seite!

**Aufgabe 3 (Berechenbarkeit)****[30 PUNKTE]**

Beantworten Sie die folgenden Fragen. Seien  $L$ ,  $L_1$  und  $L_2$  Sprachen über dem Alphabet  $\Sigma$ . Sei  $\bar{L}$  das Komplement von  $L$ , das heißt  $\bar{L} = \{w \in \Sigma^* \mid w \notin L\}$ .

- a) Definieren Sie, was man unter einer entscheidbaren Sprache versteht.
- b) Definieren Sie, was man unter einer semi-entscheidbaren Sprache versteht.
- c) Falls  $L_1$  und  $L_2$  entscheidbar sind, ist dann auch  $L_1 \cap L_2$  entscheidbar? Begründen Sie Ihre Antwort!
- d) Falls  $L$  semi-entscheidbar ist, ist dann auch  $L \cap \bar{L}$  semi-entscheidbar? Begründen Sie Ihre Antwort!
- e) Falls  $L_1$  entscheidbar ist und  $L_2$  semi-entscheidbar, ist dann auch  $L_1 \cap L_2$  entscheidbar? Begründen Sie Ihre Antwort!
- f) Falls  $L$  entscheidbar ist, ist dann auch  $\bar{L}$  entscheidbar? Begründen Sie Ihre Antwort!
- g) Falls  $L_1$  und  $L_2$  kontextfrei sind, ist dann  $L_1 \cap L_2$  entscheidbar? Begründen Sie Ihre Antwort!