

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

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

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

**Herbst  
2007**

**66115**

---

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

---

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Theoretische Informatik, Algorithmen**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

---

Thema Nr. 1

Hinweis:

Die einzelnen Teilaufgaben bauen oftmals aufeinander auf, sind aber im Prinzip in beliebiger Reihenfolge lösbar. Sie dürfen hierbei die Angaben und Ergebnisse früherer Teilaufgaben uneingeschränkt zur Lösung späterer Teilaufgaben verwenden! Außerdem dürfen Sie Tatsachen aus dem Informatik-Duden ohne weitere Begründung als bekannt voraussetzen.

Aufgabe 1:

Ordnen Sie die folgenden formalen Sprachen bestmöglich in die Chomsky-Hierarchie ein und geben Sie eine *ausreichende Begründung* an:

- a)  $L_1 = \{a^n b^n a^n \mid n \geq 1\}$
- b)  $L_2$  sei die Menge aller terminierenden Java-Programme.
- c)  $L_3$  sei die Menge aller *vollständig und korrekt geklammerten* arithmetischen Ausdrücke in den Variablen  $a$  und  $b$  mit den Operationen  $+$  und  $\times$ .  
Zur Illustration:  $((a + (b + a)) \times a) \in L_3, ((a + b + a)) \times b \notin L_3$  (nicht korrekt geklammert),  $a \times (b + b)) \notin L_3$  (nicht vollständig geklammert).
- d)  $L_4 = \{w \in \{a, b\}^* \mid w \text{ enthält mindestens 4 Vorkommen von } a\}$ .
- e)  $L_5 = \{a^n b^n \$w \mid n \geq 1 \text{ und } w \in \{a, b\}^*\}$

Aufgabe 2:

Geben Sie zu dem nichtdeterministischen endlichen Automaten in der Abbildung einen äquivalenten deterministischen Automaten an.

Ist Ihr Automat minimal?

Falls nein, so geben Sie mindestens ein Paar von Zuständen an, die zu einem einzigen Zustand zusammengefasst werden können.

Falls ja, so geben Sie für mindestens drei Zustandspaare Ihrer Wahl jeweils eine Begründung dafür, dass diese nicht zusammengefasst werden können.

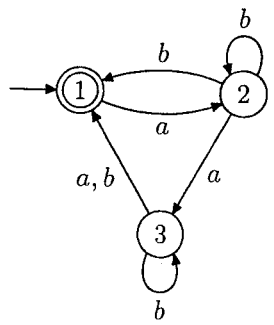


Abbildung: ein nichtdeterministischer endlicher Automat mit 3 Zuständen

**Aufgabe 3:**

Für eine (deterministische) Turingmaschine  $T = (I, \Sigma, Q, \delta, q_0, F, b)$  und ein Wort  $w \in \Sigma^*$  ist die partielle Funktion  $\text{TIME}_T(w)$  definiert als die Anzahl von Arbeitsschritten, die  $T$  bei Eingabe  $w$  ausführt. Falls  $T$  bei Eingabe  $w$  nicht hält, ist  $\text{TIME}_T(w) = \perp$ , also undefiniert. Im Folgenden sei  $\Sigma = \{0, 1\}$  fest; das leere Wort wird wie üblich mit  $\epsilon$  bezeichnet.

Die *Busy-Beaver Funktion*  $\text{BB}(n)$  ist definiert als

$$\text{BB}(n) := \max \{ \text{TIME}_T(\epsilon) \mid \text{ Turingmaschine } T \text{ hat höchstens } n \text{ Zustände und hält auf leerer Eingabe} \}$$

- Das Halteproblem bei leerer Eingabe ist die Menge  $H_0 = \{T \mid \text{TIME}_T(\epsilon) \neq \perp\}$ . Bekanntlich ist  $H_0$  unentscheidbar. Geben Sie eine Reduktion des Graphen von  $\text{BB}$ , also der Menge  $G = \{(n, b) \mid b = \text{BB}(n)\}$  auf  $H_0$  an.
- Zeigen Sie durch Widerspruch:  $\text{BB}(n)$  wächst schneller als jede berechenbare Funktion, d. h. für jede berechenbare Funktion  $f : \mathbb{N} \rightarrow \mathbb{N}$  gilt:  $\text{BB}(n) \notin O(f(n))$ .

**Aufgabe 4:**

Wrestling ist ein Showkampf, bei dem es zwei Arten von Teilnehmern gibt: Gute und böse Wrestler. Wer gut und wer böse ist, wird von den Organisatoren vorab festgelegt, die Wrestler haben sich dann entsprechend zu kleiden und zu benehmen.

Zwischen manchen Wrestlern bestehen persönliche Rivalitäten und um die Kämpfe zusätzlich anzuhetzen, ist man bestrebt, die Einteilung in Gute und Böse so vorzunehmen, dass es keine Rivalitäten zwischen zwei Guten oder zwischen zwei Bösen gibt, sondern nur zwischen „Gut“ und „Böse“.

Helfen Sie dem Management, indem Sie einen effizienten Algorithmus entwerfen, der entscheidet, ob solch eine Einteilung existiert und sie ggf. berechnet. Gegeben ist hierbei eine Menge von  $W$  Wrestlern repräsentiert durch  $\{1, \dots, W\}$  und einer Liste von  $R$  Paaren einander rivalisierender Westler. „Effizient“ bedeutet hier, dass die Laufzeit  $O(W + R)$  sein muss. Beschreiben Sie Ihre Lösung in Pseudocode oder einer Programmiersprache Ihrer Wahl.

Beispiel:

Bei drei Westlern  $\{1, 2, 3\}$  und Rivalitäten zwischen 1,2 sowie 1,3 könnte man 1 als „gut“ und 2,3 jeweils als „böse“ einteilen. Besteht zusätzlich noch eine Rivalität zwischen 2 und 3, so existiert keine Lösung.

Hinweis:

Bauen Sie Ihren Algorithmus auf einem geeigneten Verfahren zum Durchlaufen von (ungerichteten) Graphen auf.

Thema Nr. 2**Aufgabe 1:**

Gegeben sei der nichtdeterministische endliche Automat  $M$  mit dem Alphabet  $\Sigma = \{a, b\}$ , der Zustandsmenge  $\{z_0, z_1, z_2, z_3\}$ , Anfangszustand  $z_0$ , Endzustand  $\{z_3\}$  und der Überföhrungsfunktion  $\delta$  mit:

$$\delta(z_0, a) = \{z_1, z_2\},$$

$$\delta(z_1, b) = \{z_0, z_1\},$$

$$\delta(z_2, a) = \{z_2, z_3\},$$

$$\delta(z_0, b) = \delta(z_1, a) = \delta(z_2, b) = \delta(z_3, a) = \delta(z_3, b) = \emptyset$$

$L(M)$  sei die von  $M$  akzeptierte Sprache.

a) Gelten folgende Aussagen?

- i) Es gibt Zeichenreihen in  $L(M)$ , die genauso viele  $a$ 's enthalten wie  $b$ 's.
- ii) Jede Zeichenreihe in  $L(M)$ , die mindestens vier  $b$ 's enthlt, enthlt auch mindestens vier  $a$ 's.

Begründen Sie Ihre Antworten.

b) Geben Sie eine reguläre (Typ-3-)Grammatik an, die  $L(M)$  erzeugt.

c) Beschreiben Sie  $L(M)$  durch einen regulären Ausdruck.

d) Konstruieren Sie aus  $M$  mit der Potenzmengen-Konstruktion (und entsprechender Begründung) einen deterministischen endlichen Automaten, der  $L(M)$  akzeptiert.

**Aufgabe 2:**

Für beliebiges  $m \in \mathbb{N}$  sei  $L_m$  die Sprache  $L_m = \{a^i b^m a^i b^m \in \{a, b\}^* \mid i \in \mathbb{N}\}$ .

a) Beweisen Sie:  $L_3$  ist nicht regulär.

b) Ist  $L_m$  für jedes  $m \in \mathbb{N}$  nicht regulär? Begründen Sie Ihre Antwort.

c) Geben Sie die allgemeine Form einer kontextfreien (Typ-2-)Grammatik an, die  $L_m$  (für beliebiges  $m$ ) erzeugt.

d) Ist jede der Sprachen  $L_m$  mit einer deterministischen Turing-Maschine mit einer Zeitkomplexität  $O(n^2)$  entscheidbar ( $n$  ist die Länge der jeweiligen Eingabe)? Begründen Sie Ihre Antwort.

**Aufgabe 3:**

Es seien  $\Sigma$  ein Alphabet,  $L_1$  und  $L_2$  zwei Sprachen über  $\Sigma$ .  $\epsilon$  bezeichne die leere Zeichenreihe. Gelten folgende Aussagen? Begründen Sie Ihre Antworten.

a) Ist  $L_1$  kontext-sensitiv (Typ-1), so ist die Sprache

$$L = \{w \in \Sigma^* \mid w \notin L_1 \text{ und } w \neq \epsilon\}$$

entscheidbar.

- b) Wird  $L_1$  von einem linear beschränkten Automaten  $M$  mit Zustandsmenge  $Z$  und Endzustandsmenge  $E$  akzeptiert, so akzeptiert der linear beschränkte Automat  $M'$ , der aus  $M$  entsteht, wenn man  $E$  durch  $Z \setminus E$  ersetzt, die Sprache  $\Sigma^* \setminus L_1$ .
- c) Sind  $L_1$  und  $L_2$  entscheidbar, so ist auch die Sprache

$$L_1 \circ L_2 = \{w_1 w_2 \in \Sigma^* \mid w_1 \in L_1, w_2 \in L_2\}$$

entscheidbar.

- d) Ist  $L_1$  entscheidbar und  $L_2$  semi-entscheidbar, so ist die Funktion  $f : \Sigma^* \rightarrow \Sigma^*$  mit

$$f(w) = \begin{cases} \epsilon & \text{falls } w \notin L_1 \cap L_2 \\ \text{undefiniert} & \text{sonst} \end{cases}$$

berechenbar.

- e) Sind sowohl  $L_1$  als auch  $L_2$  mit einer deterministischen Turing-Maschine mit polynomieller Zeitkomplexität entscheidbar, so gilt dies auch für  $L_1 \setminus L_2$ .

#### Aufgabe 4:

- a) Beschreiben Sie kurz allgemein die Wirkungsweise des Sortier-Algorithmus Heapsort.
- b) Welche Zeitkomplexität hat Heapsort? Begründen Sie Ihre Antwort.
- c) Beschreiben Sie konkret die einzelnen Schritte, die durchgeführt werden, wenn die Zahlenfolge

13, 8, 25, 3, 9, 20, 5, 21

mit Heapsort aufsteigend sortiert wird. Geben Sie dabei die jeweiligen Heapstrukturen sowohl als Baum als auch in ihrer üblichen Darstellung als Feld an.