

---

**Prüfungsteilnehmer**

**Prüfungstermin**

**Einzelprüfungsnummer**

---

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

**Frühjahr**

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

**1999**

**46112**

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

---

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

**Fach:** **Informatik (nicht vertieft studiert)**

**Einzelprüfung:** **Grundlagen der Informatik**

**Anzahl der gestellten Themen (Aufgaben):** 1

**Anzahl der Druckseiten dieser Vorlage:** 3

Bitte wenden!

**Sämtliche Teilaufgaben sind zu bearbeiten!****Teilaufgabe 1:**

Gegeben sei das Alphabet  $\Sigma = \{0,1\}$  und die Sprache

$$L = \{10^i1^j0 \mid i \text{ und } j \text{ sind gerade und } i, j \geq 0\}.$$

- a) Geben Sie einen regulären Ausdruck mit Sprache L an.
- b) Konstruieren Sie das Übergangsdiagramm eines nichtdeterministischen, endlichen Automaten (ohne Leerübergänge), der (genau) die Sprache L akzeptiert.
- c) Konstruieren Sie das Übergangsdiagramm eines minimalen deterministischen Automaten, der (genau) die Sprache L akzeptiert.

**Teilaufgabe 2:**

Gegeben ist die folgende Grammatik zur Erzeugung arithmetischer Ausdrücke:

$$\begin{aligned} \mathcal{G} &= (\{E\}, \{a, b, +, *, (, )\}, P, E) \\ \text{mit} \\ P &= \{ E \rightarrow E+E \mid E * E \mid (E) \mid a \mid b \} . \end{aligned}$$

- a) Zeigen Sie, dass diese Grammatik ambig (d.h. nicht eindeutig) ist.
- b) Geben Sie eine eindeutige Grammatik mit derselben Sprache an, so dass die Ableitungsbäume nach der üblichen Vorrangregel “\* bindet stärker als +” gebildet werden.

Fortsetzung nächste Seite!

**Teilaufgabe 3:**

Sei  $\mathcal{A} = (S, \Sigma, \delta, s_0, S_1)$  ein deterministischer Automat mit Zustandsmenge  $S = \{p, q, r, s_0, s_1\}$  und Alphabet  $\Sigma = \{0, 1\}$ . Der Automat soll durch ein Programm in einer beliebigen imperativen Sprache simuliert werden.

a) Deklarieren Sie dazu folgende Datentypen:

- i) Einen Datentyp "TYPE State = ..." zur Repräsentation der Zustandsmenge  $S$ .
- ii) Einen Unterbereichstyp "TYPE Sigma = ..." der ganzen Zahlen zur Repräsentation des Alphabets  $\Sigma$ .
- iii) Einen Datentyp "TYPE Transition = ..." zur Repräsentation von Zustandsübergangsfunktionen (d.h.  $\delta: S \times \Sigma \rightarrow S$  soll als ein Element von "Transition" dargestellt werden können).

b) Es sei angenommen, dass  $S_1$  aus genau einem Endzustand  $s_1$  besteht. Schreiben Sie in einer imperativen Programmiersprache Ihrer Wahl eine Prozedur mit Kopf

PROCEDURE akzeptiert (delta: Transition; s0, s1: State);

die elementweise eine Folge von Ziffern liest und bestimmt, ob die Folge zur Sprache von  $\mathcal{A}$  gehört oder nicht. Das Ende der Folge wird durch die Eingabe einer von 0 oder 1 verschiedenen Ziffer gekennzeichnet.

c) Der Automat  $\mathcal{A}$  habe den Anfangszustand  $s_0 = p$ , den Endzustand  $s_1 = r$  und folgende Übergangsfunktion  $\delta$ :

$$\delta(p, 0) = r, \delta(p, 1) = q, \delta(q, 0) = r, \delta(q, 1) = p, \delta(r, 0) = q, \delta(r, 1) = r.$$

Die Prozedur "akzeptiert" von Teil b) ist in ein Programm einzubetten, so dass das Programm eine Folge von Ziffern liest und bestimmt, ob die Folge zur Sprache des obigen Automaten gehört oder nicht.