Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: ______ Kennwort: ______ Arbeitsplatz-Nr.: ______	**Frühjahr 2011**	**46113**

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
— Prüfungsaufgaben —

Fach: **Informatik**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

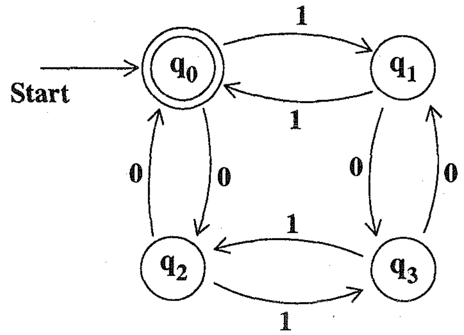
Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:
Formale Sprachen

a) Betrachten Sie den folgenden deterministischen endlichen Automaten:



Dieser Automat erkennt die Sprache

$$L_1 = \{w \mid w \in \{0,1\}^*, \ w \text{ hat gerade Anzahl von 0en und 1en } \}$$

Modifizieren Sie den Automaten so, dass er die Sprache

$$L_2 = \{w \mid w \in \{0,1\}^*, \ w \text{ hat ungerade Anzahl von 0en und 1en } \}$$

erkennt. Begründen Sie die von Ihnen vorgeschlagene Modifikation.

b) Erweitern und modifizieren Sie den gegebenen Automaten aus Aufgabenteil (a) so, dass er die Sprache

$$L_3 = \{w \mid w \in \{0,1,2\}^*, \ w \text{ hat ungerade Anzahl von 0en, 1en und 2en } \}$$

erkennt. Erläutern Sie die Arbeitsweise des von Ihnen konstruierten Automaten.

c) Beschreiben Sie, was man unter der Äquivalenz von Zuständen in deterministischen endlichen Automaten versteht. Geben Sie eine exakte Definition mit Hilfe der Übergangsfunktion an. Beschreiben Sie die Arbeitsweise des Algorithmus zur Bestimmung der äquivalenten Zustände eines deterministischen endlichen Automaten.

d) Wenden Sie den Algorithmus auf den folgenden deterministischen endlichen Automaten $M = (\{A, B, C, D, E, F, G, H\}, \{0, 1\}, \delta, A, \{D\})$ an, wobei die Übergangsfunktion δ durch folgende Tabelle gegeben ist:

Fortsetzung nächste Seite!

δ	0	1
A	B	A
B	A	C
C	D	B
D	D	A
E	D	F
F	G	E
G	F	G
H	G	D

Bauen Sie dazu die vollständige Tabelle des Algorithmus mit Zustandspaaren schrittweise auf und erläutern Sie die durchgeführten Schritte.

Fassen Sie anschließend die äquivalenten Zustände zusammen und konstruieren Sie einen äquivalenten deterministischen endlichen Automaten mit einer minimalen Anzahl von Zuständen. Geben Sie das Übergangsdiagramm des äquivalenten Automaten an.

e) Formulieren Sie die Aussage des Pumping-Lemmas für reguläre Sprachen. Wie kann das Pumping-Lemma dazu verwendet werden, zu zeigen, dass eine Sprache nicht regulär ist?

f) Beweisen Sie mit Hilfe des Pumping-Lemmas aus Aufgabenteil (e), dass die Sprache

$$L_4 = \{0^n 1 0^n 1 \mid n \geq 1\}$$

nicht regulär ist. Erklären Sie Ihre Argumentation.

g) Betrachten Sie die folgende Sprache

$$L_5 = \{a^n b^n c^m d^m \mid n \geq 1, m \geq 1\} \cup \{a^n b^m c^m d^n \mid n \geq 1, m \geq 1\}$$

Geben Sie eine kontextfreie Grammatik an, die diese Sprache beschreibt.

Aufgabe 2:
Turingmaschinen

a) Das Grundmodell einer Turingmaschine kann zur Vereinfachung der Programmierung erweitert werden, indem das Band der Turingmaschine in mehrere Spuren unterteilt wird. Erläutern Sie diese Erweiterung. Wie sieht in diesem Fall die Übergangsfunktion aus? Erklären Sie, warum eine Turingmaschine mit mehreren Spuren nicht mächtiger ist als eine Turingmaschine mit einer Spur.

b) Konstruieren Sie eine Turingmaschine mit zwei Spuren zum Erkennen der Sprache

$$L = \{wcw \mid w \in \{0,1\}^+\}$$

mit c als Markierungszeichen. Verwenden Sie dabei die 2. Spur als Markierungsspur zur Markierung von bereits gelesenen Symbolen. Geben Sie die Übergangsfunktion im Detail an. Erläutern Sie die Funktionsweise der von Ihnen konstruierten Turingmaschine.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:
Profile als formale Sprachen

Ein *Profil* ist eine endliche Folge $\boldsymbol{p} = p_1p_2 \dots p_n$ von ganzen Zahlen p_i mit $n \geq 1$, $p_1 = p_n = 0$ und $|p_{i+1} - p_i| \leq 1$ für $1 \leq i < n$. Die *Höhe* eines Profils $\boldsymbol{p}$ ist $h(\boldsymbol{p}) = \max_{1 \leq i \leq n} |p_i|$.

Als Beispiel: $(0, +1, 0, -1, -1, -2, -1, 0)$ ist ein Profil der Höhe 2. Die Folge $(0, +1, -1, -1, -2, -1, 0)$ ist kein Profil, da -1 auf $+1$ folgt. Die Folgen $(+1, 0, -1, -1, -2, -1, 0)$ bzw. $(0, +1, 0, -1, -1, -2, -1)$ sind keine Profile, da sie nicht mit 0 beginnen bzw. enden.

Die Menge P_k der Profile mit Höhe $\leq k$ kann man als eine Sprache über dem Alphabet $\Sigma_k = \{0, \pm 1, \pm 2, \dots, \pm k\}$ auffassen.

a) Zeigen Sie, dass die Sprache P_2 regulär ist, indem Sie einen endlichen Automaten angeben, der diese Sprache akzeptiert.

b) Zeigen Sie, dass die Sprache $\overline{P_2} = \Sigma_2^* \setminus P_2$ regulär ist, indem Sie einen regulären Ausdruck für $\overline{P_2}$ angeben.

c) Zeigen Sie, dass alle Sprachen P_k $(k \geq 0)$ regulär sind, indem Sie entweder die Konstruktion geeigneter endlicher Automaten für deren Akzeptanz oder die Konstruktion geeigneter Grammatiken für deren Generierung beschreiben.

Unter der *Differenz* eines Profils $\boldsymbol{p} = p_1p_2 \dots p_n$ vesteht man das Wort $\delta(\boldsymbol{p}) = w_1w_2 \dots w_{n-1} \in \Sigma_1^*$ mit $w_i = p_{i+1} - p_i$ $(1 \leq i < n)$. Die Differenz des Profils 0 ist das leere Wort.

d) Zeigen Sie, dass die Sprachen $DP_k = \{\delta(\boldsymbol{p})\,;\, \boldsymbol{p} \in P_k\} \subseteq \Sigma_1^*$ der Differenzen von Profilen regulär sind.

e) Zeigen Sie, dass für die Sprache aller Differenzen von Profilen $DP := \bigcup_{k \geq 0} DP_k \subseteq \Sigma_1^*$ gilt:

$$\forall w_1w_2 \dots w_n \in \Sigma_1^* : w_1w_2 \dots w_n \in DP \Leftrightarrow w_1 + w_2 + \dots + w_n = 0$$

und diese Sprache nicht regulär ist.

f) Zeigen Sie, dass DP eine kontextfreie Sprache ist, indem Sie entweder einen Kellerautomaten konstruieren, der die Sprache DP akzeptiert, oder indem Sie eine geeignete Grammatik für die Generierung von DP angeben.

Fortsetzung nächste Seite!

Aufgabe 2:

Das Nullfunktionsproblem

Das *Nullfunktionsproblem* stellt die Aufgabe, von beliebigen Programmen P (Turing-Maschinen, While-Programme, universelle Programmiersprache etc.) zu entscheiden, ob sie bei jeder Eingabe halten und das Resultat 0 liefern.

Es sei $P_0(z), P_1(z), P_2(z), \ldots$ eine berechenbare Auflistung aller Programme bzw. Maschinen des betrachteten Formalismus. Diese Programme haben einen Eingabeparameter z, der mit natürlichen Zahlen belegt werden kann. Die Schreibweisen $P_x(z) \downarrow$ bzw. $P_x(z) \uparrow$ sollen bedeuten, dass das Programm P_x bei Input z terminiert bzw. nicht terminiert. Bei (partiellen) Funktionen $f(z)$ bedeuten die Schreibweisen $f(z) \downarrow$ bzw. $f(z) \uparrow$, dass $f(z)$ definiert bzw. nicht definiert ist.

Es geht bei dem *Nullfunktionsproblem* um die Menge

$$D = \{\, x \in \mathbb{N} \mid \forall z \in \mathbb{N} : P_x(z) = 0\}.$$

a) Zeigen Sie unter Berufung auf die These von CHURCH, dass die Funktion

$$g(x,y) = \begin{cases} 0 & \text{falls } P_x(x) \downarrow \\ \uparrow & \text{falls } P_x(x) \uparrow \end{cases}$$

(partiell-)berechenbar ist.

b) Zeigen Sie, dass die Menge D unentscheidbar ist, indem Sie die Funktion g der vorigen Teilaufgabe für eine Reduktion verwenden.

c) Bei dem *Äquivalenzproblem* für Programme betrachtet man die Menge

$$A = \{\, (x,y) \in \mathbb{N} \times \mathbb{N} \mid \forall z \in \mathbb{N} : P_x(z) = P_y(z)\,\}.$$

Zeigen Sie mittels einer geeigneten Reduktion, dass diese Menge unentscheidbar ist.