Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: ____________	**Frühjahr**	**66112**
Kennwort: ____________	**1999**	
Arbeitsplatz-Nr.: ________		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Automatentheorie,Komplexität,Algorith.**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

Bitte wenden!

Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1:

In dieser und den folgenden Aufgaben geht es um die Klasse der sog. *regulären* Sprachen, also der Sprachen vom Typ 3 in der Klassifizierung von Chomsky.

a) Geben Sie eine genaue Definition der Klasse der regulären Sprachen an und erläutern Sie die darin vorkommenden Begriffe.

b) Es gibt mehrere äquivalente Beschreibungen für diese Klasse der regulären Sprachen (die man also ebensogut als Definitionen verwenden könnte): Zählen Sie solche weiteren Beschreibungen auf, die Sie kennen. Erläutern Sie ggf. die verwendeten Begriffe.

c) Unter welchen Operationen auf Sprachen ist die Klasse der regulären Sprachen abgeschlossen? Zählen Sie die Ihnen bekannten auf.

d) Welche Hilfsmittel kennen Sie, um die *Nicht-Regularität* einer formalen Sprache nachzuweisen. Erläutern Sie dies an einem Beispiel Ihrer Wahl.

e) In welchen Bereichen der Informatik haben reguläre Sprachen eine praktische Bedeutung? Erläutern Sie solche Anwendungen, wenn möglich unter Angabe relevanter Algorithmen (in Pseudocode).

Teilaufgabe 2:

In dieser Aufgabe werden Sprachen über dem Alphabet $\Sigma = \{0, 1\}$ betrachtet. Für ein Wort $w = w_0 w_1 \ldots w_m \in \Sigma^*$ bezeichne $(w)_2$ die Interpretation von w als Binärdarstellung einer natürlichen Zahl, wobei das führende Bit links steht:

$$(w)_2 = w_0 2^m + w_1 2^{m-1} + \ldots + w_{m-1} 2^1 + w_m 2^0$$

(N.B. es wird nicht vorausgesetzt, dass $w_0 = 1$ ist, d.h. es sind beliebig viele führende Nullen erlaubt). Für $n, k \in \mathbb{N}$ mit $0 \leq k < n$ sei

$$\mathrm{mod}(n, k) := \{w \in \Sigma^* \,;\, (w)_2 \equiv k \pmod{n}\}$$

die Sprache der Binärdarstellungen derjenigen Zahlen, die kongruent zu k modulo n sind.

a) Konstruieren Sie einen endlichen Automaten, der die Sprache $\mathrm{mod}(5, 0)$ erkennt. Wie erhält man daraus die endlichen Automaten, die die Sprachen $\mathrm{mod}(5, k)$ mit $1 \leq k \leq 4$ erkennen?

b) Formulieren Sie eine allgemeine Aussage über die Erkennbarkeit der Sprachen $\mathrm{mod}(n, k)$ durch endliche Automaten. Skizzieren Sie einen Beweis.

Fortsetzung nächste Seite!

Teilaufgabe 3:

In dieser Aufgabe werden Sprachen über dem Alphabet $\Sigma = \{a, b\}$ betrachtet. Für ein Wort $w \in \Sigma^*$ bezeichne $|w|_a$ (bzw. $|w|_b$) die Anzahl der Vorkommen von a (bzw. b) in dem Wort w. Betrachten Sie die Sprachen

$$C_k := \{w \in \Sigma^* ; \, |w|_a = |w|_b \wedge w = u \cdot v \Rightarrow \big||u|_a - |u|_b\big| \leq k\} \quad (k \geq 1)$$
$$D := \{w \in \Sigma^* ; \, |w|_a = |w|_b\}$$

Offenbar gilt

$$C_1 \subset C_2 \subset C_3 \subset \ldots \subset D \quad \text{und} \quad \lim \uparrow_{k \to \infty} C_k = D$$

a) Zeigen Sie, dass die Sprachen C_k $(k \geq 1)$ regulär sind.

b) Konstruieren Sie endliche Automaten, die die Sprachen C_1 bzw. C_2 akzeptieren.

c) Zeigen Sie, dass die Sprache D nicht regulär ist.

d) Zu welcher Sprachklasse gehört die Sprache D? Mit welchem Typ von "Automaten" kann man die Sprache D erkennen? (Auf beide Fragen ist natürlich eine möglichst stark eingegrenzte Antwort erwünscht).

Thema Nr. 2

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1:

Geben Sie *primitiv rekursive* Terme für die Funktionen in a) - e) an:

a) Die Fakultätsfunktion $f(x) = x!$.

b) Die Funktion *even* mit $even(x) = \begin{cases} 1 \text{ , wenn x gerade} \\ 0 \text{ sonst} \end{cases}$.

c) Die Signumfunktion *sig* mit $sig(x) = \begin{cases} 1 \text{ , wenn } x > 0 \\ 0 \text{ sonst} \end{cases}$.

d) Die Funktion *case* mit $case(x, y, z) = \begin{cases} y \text{ , wenn } x = 0 \\ z \text{ sonst} \end{cases}$.

e) Die ganzzahlige Division *div* mit $div(x, y) = k$ genau dann, wenn $k * y + m = x$, wobei $m < y$.

f) Gegeben sei eine primitiv rekursive Funktion $F : N_0 \to N_0$. Wir definieren nun eine Funktion $f : N_0 \to N_0$ mit $f(n) = \begin{cases} 0, \text{ falls es ein } k < n \text{ gibt mit } F(k) = 0 \\ 1 \text{ sonst.} \end{cases}$
Ist f primitiv rekursiv? Beweisen Sie Ihre Antwort. Sie dürfen voraussetzen, dass die vier Grundrechenarten (+, -, ∗, /) auf N primitiv rekursiv sind, alle weiteren von Ihnen eingeführten Hilfsfunktionen müssen explizit als primitiv rekursiv bewiesen werden.

Teilaufgabe 2:

Gegeben sei das Alphabet $\mathcal{A} = \{a, b\}$ und die Zeichenreihenmenge $K \subseteq \mathcal{A}^*$ mit $K = \{x \in \mathcal{A}^*;\ |x| \geq 2$ und das vorletzte Zeichen von x ist ein $a\}$.

a) Beschreiben Sie K durch eine reguläre Menge.

b) Geben Sie eine Typ-3-Grammatik G mit dem Sprachschatz K an und zeigen Sie, dass $\mathcal{L}(G) = K$ gilt.

c) Geben Sie einen (evtl. nichtdeterministischen) endlichen Automaten M mit $T(M) = K$ an.

d) Geben Sie einen deterministischen Kellerautomaten M an mit $N(M) = \{a^{2n}b^n;\ n \in N\}$.

Fortsetzung nächste Seite!

Teilaufgabe 3:

Programmieren Sie in einer imperativen Programmiersprache Ihrer Wahl nach folgenden Angaben:

Schreiben Sie eine Funktion *Tag*, die bestimmt, der wievielte Tag nach dem letzten Februartag eines Jahres zwischen 1900 und 2099 Ostersonntag ist. Dies lässt sich mit dem folgenden Verfahren ermitteln:

- q sei das Ergebnis der ganzzahligen Division der Jahreszahl j $(1900 < j < 2099)$ durch 4.
- a sei der Rest der ganzzahligen Division von j durch 19.
- b sei der Rest der ganzzahligen Division von $(204 - 11 * a)$ durch 30. Falls $b = 28$ oder $b = 29$ ist, so ist mit $b = 27$ bzw. $b = 28$ weiterzurechnen.
- c sei der Rest der ganzzahligen Division von $(j + q + b - 13)$ durch 7.

Das Ergebnis ist dann $28 + b - c$. Schreiben Sie ein Programm, das eine natürliche Zahl j mit $1900 < j < 2099$ einliest, nach Überprüfung auf korrekte Eingabe mit Hilfe der Funktion *Tag* und einer Prozedur *Datum* das Datum des Ostersonntags des Jahres bestimmt und das Ergebnis ausgibt. Das Datum sei dabei in der Prozedur dargestellt durch zwei Ausgabeparameter t und m (z.B. $t = 7$ und $m = 3$ für den 7. März).

Teilaufgabe 4:

Ein Firmengebäude hat 140 Räume. Es gibt ein Verzeichnis aller Räume, in dem für jeden Raum folgende Angaben gemacht werden:

- Raumkennung, bestehend aus einem Zeichen (z.B. 'E' für Erdgeschoss, 'K' für Keller) und einer natürlichen Zahl,
- Größe in Quadratmetern (als `REAL`-Zahl),
- Angabe, ob der Raum einen Telefonanschluss hat,
- Maximale Anzahl der Personen, die in diesem Raum arbeiten können (Null, wenn es sich um einen Lagerraum handelt),
- Aktuelle Anzahl der Personen, die in diesem Raum arbeiten.

Programmieren Sie in einer imperativen Programmiersprache Ihrer Wahl nach folgenden Angaben:

a) Geben Sie geeignete Datenstrukturen <u>`Raum`</u> und <u>`Raumverzeichnis`</u> an.

b) Geben Sie eine Prozedur an, die das Raumverzeichnis verändert, wenn einem neuen Mitarbeiter ein Raum zugeteilt wird, wobei man in einem Parameter angeben kann, ob man einen Raum mit oder ohne Telefonanschluss haben will.

   Beachten Sie dabei, dass die maximale Personenanzahl pro Raum nicht überschritten wird. Der erste passende Raum kann gewählt werden. Sie können annehmen, dass ein solcher Raum vorhanden ist.

c) Geben Sie eine Funktion an, die die Gesamtgröße aller Lagerräume in Quadratmetern ermittelt.