Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: ______________ Kennwort: ______________ Arbeitsplatz-Nr.: ________	**Herbst 2004**	**66112**

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: 7

Bitte wenden!

Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten !

Hinweis: Verwenden Sie zur Formulierung von Algorithmen bzw. Datentypen eine gängige höhere Programmiersprache oder einen entsprechenden Pseudocode. Erläutern Sie alle anzugebenden Algorithmen, Automaten und Turing-Maschinen ausgiebig durch Kommentare.

Teilaufgabe 1

Jede *Karte* in einem Kartenspiel hat eine *Farbe* Kreuz, Pik, Herz oder Karo und einen *Wert*. Mögliche Werte sind die *Zahlenwerte* 2 bis 10, die *Bildwerte* Bube, Dame, König sowie der Wert As. Eine *Kartenverteilung* besteht aus 12 Karten.

Geben Sie einen geeigneten Datentyp zur Darstellung von Kartenverteilungen sowie Algorithmen an, die für eine derartige Darstellung Folgendes leisten:

a) Für eine Kartenverteilung v und eine Karte k soll festgestellt werden, ob k in v enthalten ist.

b) Für eine Kartenverteilung v soll die Anzahl der in v enthaltenen Karten k mit folgender Eigenschaft bestimmt werden: k hat einen Bildwert, und v enthält kein As von gleicher Farbe.

Teilaufgabe 2

Gegeben sei die Funktionsvereinbarung (in Pseudocode-Notation)

```
function f(x,y: integer) integer :
    if x < -y then x - y else f(x + 2, y - 3) endif
```

a) Bestimmen Sie den Wert von $f(-5,8)$.
b) Beweisen Sie: f terminiert für alle **integer**-Zahlen x, y.
c) Geben Sie einen iterativen Algorithmus an, der $f(x, y)$ für beliebige **integer**-Zahlen x, y berechnet.

Teilaufgabe 3

Gegeben sei das Alphabet $\sum = \{a, b\}$ und die Grammatik $G = (\{S, A, B\}, \sum, P, S)$ mit der Menge P der Regeln

Fortsetzung nächste Seite!

S → aB
B → b | bA
A → aB

£(G) sei die von G erzeugte Sprache.

a) Geben Sie eine Ableitung von *ababab* in G an.

b) Beweisen Sie: £(G) = $\left\{(ab)^n \in \sum^* \mid n \geq 1\right\}$.

c) Konstruieren Sie direkt aus G zunächst einen nichtdeterministischen endlichen Automaten, der die Sprache *£(G)* akzeptiert, und daraus einen deterministischen endlichen Automaten, der ebenfalls *£(G)* akzeptiert.

d) Wieviele Zustände muss jeder deterministische endliche Automat, der *£(G)* akzeptiert, mindestens haben? Begründen Sie Ihre Antwort.

e) Ersetzen Sie die Regel $A \rightarrow aB$ in G durch eine kontextfreie Regel der Form $A \rightarrow x$ mit $x \in \{S,A,B,a,b\}^*$, so dass die derart veränderte Grammatik die Sprache

$$\left\{(ab)^{2n+1} \in \sum^* \mid n \geq 0\right\}$$

erzeugt. Begründen Sie Ihre Antwort.

f) Die Sprachen L_1 und L_2 seien gegeben durch

$$L_1 = \left\{(ab)^n (ab)^n \in \sum^* \mid n \geq 1\right\}, L_2 = \left\{(ab)^n (ba)^n \in \sum^* \mid n \geq 1\right\}.$$

i) Ist L_1 regulär?
ii) Ist L_1 kontextfrei?
iii) Ist L_2 regulär?
iv) Ist L_2 kontextfrei?

Begründen Sie Ihre Antworten.

Fortsetzung nächste Seite!

Teilaufgabe 4

a) Definieren Sie in Pseudocode oder einer höheren Programmiersprache eine geeignete Datenstruktur zur Repräsentation von AVL-Bäumen.
Schreiben Sie sodann eine Funktion, die überprüft, ob ein beliebiges Element dieser Datenstruktur tatsächlich ein AVL-Baum ist. Zu Ihrer Information: Solch eine Funktion kann zu Testzwecken sinnvoll sein, spielt aber bei der regulären Verwendung von AVL-Bäumen keine Rolle.
Beachten Sie, dass ein AVL-Baum insbesondere auch ein binärer Suchbaum ist.

b) Schreiben Sie eine Funktion, die eine Linksrotation der Wurzel eines AVL Baumes durchführt. Beschreiben Sie detailliert, welche Form der Balancierungsstörung durch diese Rotation behoben werden kann und in welchen Situationen (Einfügen, Löschen) es zu ihr kommen kann. Geben Sie die Laufzeit ihrer Implementierung mit Hilfe der O-Notation an.

Thema Nr. 2

Sämtliche Teilaufgaben sind zu bearbeiten !

1. Teilaufgabe (Automatentheorie)

Es sei E die Menge aller Wörter über dem Alphabet $\{0, 1\}$, die mit 01 enden.
a) Geben Sie einen regulären Ausdruck an, der E beschreibt.
b) Geben Sie einen deterministischen endlichen Automaten an, der E akzeptiert.
c) Beschreiben Sie die natürlichen Zahlen, deren Binärdarstellung in E liegt, auf andere Weise.

2. Teilaufgabe (Formale Sprachen)

Geben Sie eine kontextfreie Grammatik an, die die Sprache $A = \text{def} \left\{ uvv^R ww^R u^R : u, v, w \in \{a, b\}^* \right\}$ erzeugt. Dabei ist $(a_1 a_2 \ldots a_n)^R =_{\text{def}} a_n \ldots a_2 a_1$ für alle $a_1, a_2, \ldots, a_n \in \{a, b\}$.

3. Teilaufgabe (Berechenbarkeit)

Man gebe eine Turingmaschine mit einem Band an, die die Funktion $f : \{0,1\}^* \to \{0,1\}^*$ definiert durch:

$$f(w) =_{\text{def}} \begin{cases} 1, \text{ falls die Anzahl der Einsen in } w \text{ durch 3 teilbar ist} \\ 0 \text{ sonst} \end{cases}$$

berechnet.

4. Teilaufgabe (Komplexität)

Geben Sie einen nichtdeterministischen Algorithmus an, der das Partitionsproblem

$$\text{PARTITION} =_{\text{def}} \left\{ (\mathrm{a}_1, \ldots, a_m) : m, a_1, \ldots, a_m \in \mathbb{N} \wedge \exists I \left(I \subseteq \{1, \ldots, m\} \wedge \sum_{i \in I} a_i = \sum_{i \notin I} a_i \right) \right\}$$

in Polynomialzeit löst. Analysieren Sie die Laufzeit Ihres Algorithmus.

Fortsetzung nächste Seite!

5. Teilaufgabe (objektorientierter Entwurf)

Ein Volkslauf besitzt einen Namen, die Länge der Strecke, der Ort und das Datum sind weitere wichtige Eigenschaften. Ein Volkslauf wird von Mitgliedern einer Laufsportgruppe veranstaltet. Läufer, die an dem Lauf teilnehmen wollen, müssen zur Anmeldung Ihren Namen, die Laufsportgruppe, der sie angehören und ihre Altersklasse mitteilen. Es soll deutlich zwischen der Rolle als Laufveranstalter und als Mitläufer unterschieden werden.

a) Modellieren Sie den Sachverhalt durch ein UML Klassendiagramm. Dabei soll zum Ausdruck kommen, dass an einem Volkslauf mindestens 50 Läufer teilnehmen und ein Läufer durchaus an mehreren Volksläufen teilnehmen kann.

b) Geben Sie ein Objektdiagramm für folgende Situation

- Sammy nimmt teil an Residenzlauf.
- Sammy nimmt teil an Nikolauslauf.
- Hans veranstaltet Nikolauslauf.
- Paul nimmt teil an Residenzlauf.
- Haile nimmt teil an Residenzlauf.

c) Implementieren Sie die Klasse Läufer in Java. Dabei soll ein Läufer einen Namen, eine Laufsportgruppe und sein Geburtsjahr als Attribute besitzen. Eine Methode bestimmt die Altersklasse. Altersklassen sind nur vom Geburtsjahr abhängig.

- unter 18 Jahre : Jugend
- 19 bis 29 Jahre : Haupt
- zwischen i und $i + 4$ Jahren : *Mi* $i = 30,35,...$

d) Sehen Sie nun vor, dass der Läufer auch seine Volkslauf-Teilnahmen einsehen kann. Legen Sie eine entsprechende Datenstruktur an.
Schreiben Sie eine Methode
`printLaeufe (int  jahr),`

die die Namen aller Läufe, bei denen er im Jahr `jahr` mitgelaufen ist, ausgibt.

6. Teilaufgabe (Systemmodellierung)

Ein Kaffeeautomat wird durch 3 Knöpfe bedient. Diese können nur gedrückt werden, falls keine Lampe blinkt.

Nach dem Einschalten mit Knopf 1 wird das Wasser aufgeheizt, Lampe 1 blinkt. Ist dieser Vorgang beendet, so leuchtet Lampe 1 und durch Druck auf Knopf 2 kann das Mahlen und Ausgießen des Kaffees erreicht werden. Während dieses Vorganges blinkt Lampe 2. Durch Druck auf Knopf 3 wird Dampf bereitet, Lampe 3 blinkt. Ist diese Aufbereitung fertig (Lampe 3 leuchtet nun), so kann durch Knopf 2 Dampf abgelassen werden. Während dieses Vorganges blinkt Lampe 2. Durch erneuten Druck auf Knopf 3 kann zur Kaffee-Ausgabe zurückgeschaltet werden, wieder blinkt Lampe 3. Knopf 1 schaltet in jedem Modus den Automat aus.

Modellieren Sie diesen Sachverhalt durch einen Zustandsautomaten.

Fortsetzung nächste Seite!

7. Teilaufgabe (Algorithmen und Datenstrukturen)

Ein gerichteter Distanzgraph sei durch seine Adjazenzmatrix gegeben. (In einer Zeile stehen die Längen der von dem Zeilenkopf ausgehenden Wege.)

	M	A	P	R	N
M	-	5	10	-	-
A	-	-	3	9	2
P	-	2	-	1	-
R	-	-	-	-	4
N	7	-	-	6	-

a) Stellen Sie den Graph in der üblichen Form dar.
b) Bestimmen Sie mit dem Algorithmus von Dijkstra ausgehend von M kürzeste Wege zu allen anderen Knoten.
c) Beschreiben Sie wie ein Heap als Prioritätswarteschlange in diesem Algorithmus verwendet werden kann.
d) Geben Sie die Operation: „entfernen des Minimums" für einen Heap an. Dazu gehört selbstverständlich die Restrukturierung des Heaps.