Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr
2012

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: 9

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

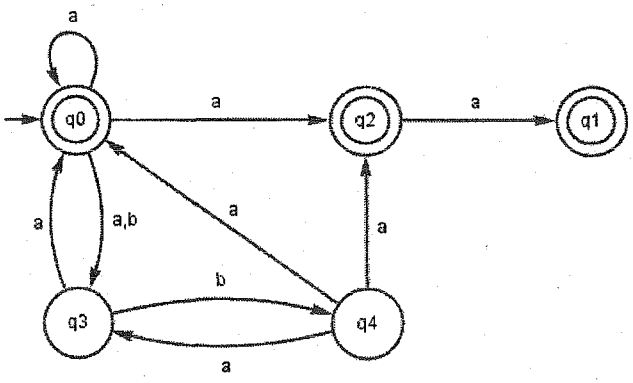
Aufgabe 1:

Sei $M_0, M_1, \ldots$ eine Gödelisierung der Registermaschinen (Random-Access-Maschinen, RAMs). Bestimmen Sie, ob folgende Mengen aufzählbar sind. Bestimmen Sie außerdem, ob die Mengen entscheidbar sind. Beweisen Sie Ihre Aussagen!

$$A_1 = \{(i,t) \in \mathbb{N} \times \mathbb{N} \mid M_i \text{ hält auf Eingabe } i \text{ nicht innerhalb von } t \text{ Rechenschritten}\}$$
$$A_2 = \{(i,x) \in \mathbb{N} \times \mathbb{N} \mid M_i \text{ hält nicht auf Eingabe } x\}$$
$$A_3 = \{i \in \mathbb{N} \mid \text{für die von } M_i \text{ berechnete Funktion } f : \mathbb{N} \to \mathbb{N} \text{ gilt } \exists x \in \mathbb{N}, f(x) = x\}$$

Aufgabe 2:

Gegeben ist der folgende nichtdeterministische, endliche Automat M. Konstruieren Sie einen endlichen Automaten M' mit $L(M') = \{a,b\}^* \setminus L(M)$.



Aufgabe 3:

Beweisen Sie, dass folgende Sprache kontextfrei, aber nicht regulär ist.

$$C = \{a^n b^m \mid n \geq m \geq 1\}$$

Aufgabe 4:

Gegeben ist die kontextfreie Grammatik $G = (\Sigma, N, S, R)$ mit $\Sigma = \{a,b\}$, $N = \{S, A, B\}$ und $R = \{S \to A, S \to B, A \to aAb, B \to AA, B \to bBa, A \to a\}$. Geben Sie eine äquivalente Grammatik in Chomsky-Normalform an.

Fortsetzung nächste Seite!

Aufgabe 5:

Gegeben sind zwei Varianten des Teilsummenproblems (sum of subset, SOS). Beweisen Sie: S_1 ist auf S_2 in Polynomialzeit reduzierbar.

$$S_1 = \{(a_1, \ldots, a_m, c) \mid m, a_1, \ldots, a_m, c \in \mathbb{N} \text{ und } \exists b_1, \ldots, b_m \in \{0, 1, 2\} \text{ mit } \sum_{i=1}^{m} b_i \cdot a_i = c\}$$

$$S_2 = \{(a_1, \ldots, a_m, c) \mid m, a_1, \ldots, a_m, c \in \mathbb{N} \text{ und } \exists I \subseteq \{1, \ldots, m\} \text{ mit } \sum_{i \in I} a_i = c\}$$

Aufgabe 6:

Binäre Suchbäume

a) Gegeben sei eine Folge von n Zahlen, über deren Verteilung nichts bekannt ist. Es soll ein binärer Suchbaum konstruiert werden, der die Zahlen in dieser Folge speichert. Argumentieren Sie, warum es dafür keinen Algorithmus mit linearer Worst-Case-Laufzeit geben kann. (Um für zwei Zahlen a und b zu entscheiden, ob $a \leq b$, gehen wir davon aus, dass eine Methode compare(a, b) verwendet werden muss. Diese liefert in $O(1)$ Zeit true, falls $a \leq b$ und sonst false.)

b) Angenommen die Zahlenfolge sei aufsteigend sortiert. Geben Sie nun einen Linearzeitalgorithmus an, der die Zahlen in dieser Folge in einem binären Suchbaum speichert.

c) Erweitern oder modifizieren Sie Ihren Algorithmus so, dass er in Linearzeit einen *balancierten* Binärbaum, also etwa einen Rot-Schwarz-Baum oder einen AVL-Baum ausgibt, der die Zahlen in der gegebenen Folge enthält.

Zeigen Sie, dass der von Ihrem Algorithmus konstruierte Baum tatsächlich die Eigenschaften eines Rot-Schwarz- oder AVL-Baums besitzt.

Fortsetzung nächste Seite!

Aufgabe 7:
Kürzeste Kreise

Mit der *Länge* eines Pfads oder eines Kreises bezeichnen wir die Anzahl der Kanten, aus denen der Pfad bzw. der Kreis besteht. Bekanntlich kann man Breitensuche verwenden, um für zwei gegebene Knoten s und t die Länge eines kürzesten s-t-Wegs zu berechnen. Im folgenden geht es um die Berechnung kürzester Kreise.

a) Für einen Graphen G und einen Knoten v von G berechnet $\mathsf{KK}(G, v)$ (siehe Abbildung 1) die Länge des kürzesten Kreises in G, der durch v geht.

   Analysieren Sie die Laufzeit von KK in Abhängigkeit von der Anzahl n der Knoten von G, von der Anzahl m der Kanten von G und vom Grad $\deg(v)$ des übergebenen Knotens v.

b) Wenn man den Algorithmus KK für jeden Knoten eines Graphen G aufruft, kann man die Länge eines kürzesten Kreises in G berechnen. Welche Laufzeit hat der resultierende Algorithmus in Abhängigkeit von n und m?

c) Geben Sie einen Algorithmus $\mathsf{KKschnell}(G, v)$ an, der in $O(n+m)$ Zeit die Länge des kürzesten Kreises in G berechnet, der durch v geht.

   Argumentieren Sie, warum ihr Algorithmus korrekt ist.

Abbildung 1

KK(ungerichteter ungewichteter Graph $G = (V, E)$, Knoten v)

```
1 L := ∞
2 Adj[v] := {w ∈ V | {v, w} ∈ E}
3 foreach w ∈ Adj[v] do
4 |   Sei G' der Graph G ohne die Kante {v, w}.
5 |   Sei ℓ die Länge eines kürzesten v-w-Wegs in G'.
6 |   if ℓ < L then
7 |   |   L := ℓ
8 return L
```

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:

Für natürliche Zahlen a, b mit $b > 0$ bezeichne $\texttt{div}(a, b)$ den Quotienten und $\texttt{mod}(a, b)$ den Rest der ganzzahligen Division von a durch b, d.h. es gilt

$$a = b \cdot \texttt{div}(a, b) + \texttt{mod}(a, b) \quad \text{mit} \quad 0 \leq \texttt{mod}(a, b) < b,$$

und die Divisionseigenschaft der ganzen Zahlen besagt, dass $\texttt{div}(a, b)$ und $\texttt{mod}(a, b)$ durch diese Beziehung eindeutig bestimmt sind. Für $b = 0$ setzen wir div(a,b) = mod(a,b) = 0 fest.

Mit $\texttt{ggT}(a, b)$ wird der grösste gemeinsame Teiler von a und b bezeichnet, wobei $\texttt{ggT}(a, 0) = a$ ist (vereinbarungsgemäß auch für den Fall $a = 0$).

a) Zeigen Sie, dass `mod` eine primitiv-rekursive Funktion ist.

b) Zeigen Sie, dass `div` eine primitiv-rekursive Funktion ist.

c) Geben Sie ein LOOP-Programm zur Berechung von ggT an.

Aufgabe 2:

Welche folgenden Behauptungen über Sprachen $A, B \subseteq \Sigma^*$ sind korrekt und welche sind falsch. Begründen Sie Ihre Antwort!
$A \setminus B$ bezeichnet die Mengendifferenz.

a) Sind A und B regulär, so ist auch $A \setminus B$ regulär.

b) Sind A und B kontextfrei, so ist auch $A \setminus B$ kontextfrei.

c) Sind A und B entscheidbar, so ist auch $A \setminus B$ entscheidbar.

d) Sind A und B partiell-entscheidbar, so ist auch $A \setminus B$ partiell-entscheidbar.

e) Sind A und B polynomiell-entscheidbar, d.h. in der Komplexitätsklasse P, so ist auch $A \setminus B$ polynomiell-entscheidbar.

Fortsetzung nächste Seite!

Aufgabe 3:

Es sei Σ ein endliches Alphabet. Die Menge Γ bestehe aus allen kontextfreien Grammatiken mit Terminalalphabet Σ. Das Problem $L \subseteq \Gamma \times \Sigma^*$ bestehe aus allen Paaren $(G, w) \in \Gamma \times \Sigma^*$ mit $G \vdash^* w$, d. h. w ist mit Hilfe von G ableitbar.

IS bezeichne das Problem INDEPENDENTSET - ein bekanntes NP-vollständiges Problem.

Kommentieren Sie die beiden folgenden Reduktions-Behauptungen unter der Hypothese, dass $\mathsf{P} \neq \mathsf{NP}$ gilt:

1. $IS \leq_p L$,
2. $L \leq_p IS$.

Welche der Behauptungen treffen zu bzw. treffen nicht zu? Mit Begründung! Hinweis: In welcher Komplexitätsklasse liegt L?

Fortsetzung nächste Seite!

Aufgabe 4:

Die drei Symbolmengen

$$\Sigma_0 = \{a, b, c, d, e, f\}, \Sigma_1 = \{-\}, \Sigma_2 = \{+, *, /\}$$

ergeben das Alphabet $\Sigma = \Sigma_0 \cup \Sigma_1 \cup \Sigma_2$. Die Elemente von Σ_i stellen i-stellige Funktionssymbole dar ($i \in \{0, 1, 2\}$). Die nullstelligen Funktionssymbole a, b, c, d, e, f sind symbolische Konstanten.

Die Menge der *Terme* `Te` ist folgendermaßen definiert:

- jedes $x \in \Sigma_0$ gehört zu `Te`;
- für jedes $y \in \Sigma_1$ und jeden Term $\mathtt{t} \in \mathtt{Te}$ gehört $y\,\mathtt{t}$ zu `Te`;
- für jedes $z \in \Sigma_2$ und je zwei Terme $\mathtt{s}, \mathtt{t} \in \mathtt{Te}$ gehört $z\,\mathtt{s}\,\mathtt{t}$ zu `Te`.

a) Geben Sie eine kontextfreie Grammatik G an, von der die Menge der Terme generiert wird: $\mathcal{L}(G) = \mathtt{Te}$.

b) Geben Sie einen Term $\mathtt{xsol} \in \mathtt{Te}$ an, der die x-Komponente der Lösung eines linearen Gleichungssystems

$$\begin{aligned} ax + by &= e \\ cx + dy &= f \end{aligned}$$

darstellt. Zeichen aus $\Sigma_1 \cup \Sigma_2$ haben ihre übliche Bedeutung.

Zur Erinnerung: für eine Lösung (x, y) gilt in üblicher Notation

$$\begin{aligned} x &= \frac{ed - bf}{ad - bc} \\ y &= \frac{af - eb}{ad - bc} \end{aligned}$$

Fortsetzung nächste Seite!

c) Zeigen Sie $G \vdash^* \texttt{xsol}$ durch Angabe einer Ableitung.

d) Eine Funktion $h : \Sigma^* \to \mathbb{Z}$ wird eindeutig definiert durch:

- $h(\epsilon) = 0$
- $\forall x \in \Sigma_i : h(x) = i - 1 \ (i \in \{0, 1, 2\})$
- $\forall u, v \in \Sigma^+ : h(u\,v) = h(u) + h(v)$

Beispiele:

$\texttt{s} = +a - a \in L$	$\texttt{t} = +a + b \notin L$	$\texttt{u} = /a * a - b \in L$	$\texttt{v} = +a - a * b \notin L$
$h(\texttt{s}) = -1$	$h(\texttt{t}) = 0$	$h(\texttt{u}) = -1$	$h(\texttt{v}) = -1$

Beweisen Sie die Richtung $\Rightarrow$ in der folgenden Äquivalenz:

$$(\mathcal{L}) \quad \forall w \in \Sigma^* : w \in \texttt{Te} \Leftrightarrow \begin{cases} h(w) = -1 \quad \text{und} \\ \forall u, v : w = u\,v \wedge v \neq \epsilon \Rightarrow h(u) \geq 0 \end{cases}$$

e) Verwenden Sie das Kriterium $(\mathcal{L})$, um einen Kellerautomaten zu konstruieren, der die Sprache Te akzeptiert.

Fortsetzung nächste Seite!

Aufgabe 5:

Folgendes Codefragment stellt einen Sortieralgorithmus dar:

```
1       static void sortiere(int p, int q, int[] a) {
2           int m,x,y,t,i;
3           if (p>=q) {System.out.print("Fehler!");}
4           m=a[(p+q)/2];
5           x=p; y=q;

6           do {
7               while(a[x]<m) {x++;}
8               while(a[y]>m) {y--;}
9               if(x<y) {
10                  t=a[x];
11                  a[x]=a[y];
12                  a[y]=t;
13                  x++; y--;}
14              else break;}
15          while (x<=y);

16          if (x==y) {x++; y--;}
17          if (y>p) {sortiere(p,y,a);}
18          if (x<q) {sortiere(x,q,a);}}
```

a) Tragen Sie in einer Tabelle (s. u.) ein, wie oft jede der Programmzeilen 7, 8, 16, 17 und 18 im besten sowie im schlechtesten Fall ausgeführt wird. Nehmen Sie an, dass Variablen korrekt vereinbart wurden. Das zu sortierende Feld enthält 5 Elemente.

Programmzeile	bester Fall	schlechtester Fall
7	v-mal	w-mal

b) Geben Sie jeden Zwischenschritt bis zur sortierten Liste für folgenden Aufruf an:

sortiere(0, 4, [4, 6, 3, 1, 2])

c) Geben Sie eine worst-case Abschätzung des Aufwandes der Methode *sortiere* in O-Notation an.

d) Bei obigem Algorithmus wird die Zahlenliste im Datentyp Feld gespeichert. Alternativ könnte auch eine einfach verkettete Liste verwendet werden. Vergleichen Sie in maximal drei Sätzen den Einsatz dieser beiden Datentypen bei der Verwendung des obigen Sortierverfahrens hinsichtlich Speicherbedarf und Gesamtlaufzeit.