Sammlung aller Staatsexamensaufgaben der Prüfungsnummer

46112

Grundlagen der Informatik (nicht vertieft)

46112

Grundlagen der Informatik (nicht vertieft)

Frühjahr 1990

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

FRÜHJAHR 1990

46112

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: 4

Bitte wenden!

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1

Gegeben sei die folgende Funktionsprozedur in PASCAL-Notation:

```
FUNCTION f(n : NAT) : NAT;
        BEGIN IF n <= 1 THEN f := 1
              ELSE f := f(n-1) + 2 * f(n-2)
        END;
```

Dabei sei die Art NAT durch TYPE $NAT = \{INTEGER\ x : x >= 0\}$ definiert.

1.1 Zeigen Sie, daß für $n >= 0$ gilt:

$$f(n) = \frac{2^{n+1} - (-1)^{n+1}}{3}$$

1.2

a) Schreiben Sie eine rekursive Funktionsprozedur für eine Funktion

$$a : NAT \to NAT,$$

die folgender Spezifikation genügt:

$a(n)$ gibt die Gesamtzahl der Aufrufe von f für $f(n)$ an!

b) Vergleichen Sie die Folge $A = \{a(n)\}_{n=0,1,2,3,4,\ldots}$ mit der Fibonaccifolge $F = \{1, 1, 2, 3, 5, \ldots\}$ und beweisen Sie, daß A eine Majorante von F ist!

1.3 Schreiben Sie eine rekursive Funktionsprozedur

$$\boldsymbol{q : NAT \to NAT}$$

für die Funktion $q(n) = 2^n$. Die Aufrufkomplexität von q soll O(log n) sein!

1.4 Schreiben Sie unter Verwendung der Funktion q (aus Teilaufgabe 1.3) eine Funktionsprozedur für eine Funktion

$$g : NAT \to NAT,$$

die sich nicht selbst aufruft und den gleichen Wertverlauf wie f hat!

Fortsetzung nächste Seite!

Teilaufgabe 2

Gegeben seien die folgenden Arten (in PASCAL-Notation):

```
TYPE name   = ARRAY[1..20] OF CHAR;
     datum  = 18000101..19991231;
     liste  = ↑ eintrag;
     eintrag= RECORD n : name;
                     gebtag : datum;
                     next : liste
              END;
```

Hierbei stellt ein Objekt der Art *datum* als 8-stellige Zahl $j_1j_2j_3j_4m_1m_2t_1t_2$ das Datum $t_1t_2.m_1m_2.j_1j_2j_3j_4$ dar; z.B. steht 19880917 für das Datum 17.9.1988. Zu beachten ist allerdings, daß nicht jedes Objekt der Art *datum* sinnvoll interpretiert werden kann; z.B. sind 18771405, 19010547 und 19890229 nicht interpretierbar, weil es die entsprechenden Daten nicht als Tagesdatum gibt.

Sei nun *personal* eine Variable der Art *liste*. Die Variable *personal* verweise als Anker auf eine nicht-zyklische, einfach verkettete Liste von Objekten der Art *eintrag*. Entwickeln Sie eine geschlossene Rechenstruktur, die es erlaubt, einen sortierten Binärbaum als Zugriffsindex, nach Geburtsdaten geordnet, auf die mit *personal* bezeichnete Liste aufzubauen! Die Personalliste soll dabei unverändert bleiben, und es darf nicht angenommen werden, daß sie geordnet wäre. Der sortierte Binärbaum als Zugriffsindex soll keinen Namen aus der Personaldatei enthalten. Im einzelnen ist für diese Aufgabe folgendes zu leisten:

2.1 Legen Sie die Datenstruktur für den aufzubauenden sortierten Binärbaum fest!

2.2 Schreiben Sie eine Prozedur *einf*, die ein Element in den sortierten Binärbaum so einträgt, daß die Zugriffsbedingung (geordnet nach Geburtsdaten) auf *personal* beachtet wird!

2.3 Schreiben Sie eine Prozedur *aufbau*, die unter Verwendung von *einf* (aus Teilaufgabe 2.2) einen sortierten Binärbaum aufbaut. Dieser Baum soll der Aufgabe entsprechend als Zugriffsindex für die gegebene, mit *personal* bezeichnete Liste verwendet werden!

2.4 Schreiben Sie eine Prozedur *gibaus*, die mit Hilfe des aufgebauten Binärbaums alle Einträge aus der Personalliste zu einem vorgegebenen Geburtsdatum heraussucht und die zugehörigen Namen ausdruckt! Falls die Liste keinen Eintrag mit dem gegebenen Geburtsdatum enthält, soll der Hinweis 'kein Eintrag' ausgedruckt werden.

Fortsetzung nächste Seite!

Teilaufgabe 3

Gegeben sei das Alphabet $A = (a, b, c, x, y)$.
Für eine Sprache S über A seien die relativen Häufigkeiten, mit denen die Zeichen aus A auftreten, durch folgende Tabelle gegeben:

a	41%
b	19%
c	10%
x	23%
y	7%

Betrachtet wird die folgende Binärcodierung C_1 für A:

$$C_1 \triangleq \begin{pmatrix} \text{a:} & O \\ \text{b:} & LOL \\ \text{c:} & LLO \\ \text{x:} & LOO \\ \text{y:} & LLL \end{pmatrix}$$

3.1 Bestimmen Sie die mittlere Codewortlänge von C_1 für die oben angegebenen relativen Häufigkeiten!

3.2 C_1 erfüllt die Fano-Bedingung und ist daher eindeutig decodierbar. Geben Sie unter Verzicht auf die eindeutige Decodierbarkeit eine Binärcodierung C_2 für A an, die hinsichtlich der mittleren Codewortlänge optimal ist!

3.3 Bestimmen Sie die mittlere Codewortlänge von C_2!

3.4 Die Codierung soll hinsichtlich der mittleren Codewortlänge gegenüber C_2 durch folgende Maßnahme weiter verbessert werden:
Das Buchstabenpaar aa soll durch ein eigenes Codewort dargestellt werden, d.h. betrachtet wird das Alphabet $A' = (a_1, a_2, b, c, x, y)$ mit folgender Maßgabe:
a_1 steht für a und a_2 für ein Paar aa in unmittelbarer Folge. Jede Zeichenreihe z über A läßt sich dann offenbar (wenn auch nicht eindeutig) durch eine Zeichenreihe z' über A' so darstellen, daß die ursprüngliche Zeichenreihe z eindeutig wiedergewonnen werden kann und in z' kein Paar a_1a_1 in unmittelbarer Folge auftritt.
z.B. ist $z = xabbaaaaac$ durch
$z'_1 = xa_1bba_2a_2a_1c$, aber auch durch
$z'_2 = xa_1bba_2a_1a_2c$ und durch
$z'_3 = xa_1bba_1a_2a_2c$
darstellbar.

3.4.1 Bestimmen Sie die relativen Häufigkeiten für die Zeichen von A' entsprechend der oben angegebenen Tabelle!

3.4.2 Geben Sie eine hinsichtlich der mittleren Codewortlänge optimale Binärcodierung für A' unter Verzicht auf eindeutige Decodierbarkeit an!

3.4.3 Bestimmen Sie die mittlere Codewortlänge von A' sowie die mittlere Codewortlänge für die dadurch gegebene Binärcodierung von A!

46112

Grundlagen der Informatik (nicht vertieft)

Frühjahr 1995

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr 1995

46112

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: 2

Bitte wenden!

Sämtliche Teilaufgaben sind zu bearbeiten!

1) Welche Eigenschaften charakterisieren einen Algorithmus?

2) Beschreiben Sie eine universelle Turingmaschine möglichst exakt.

3) a) Wie kann man das Terminieren eines Programms beweisen?

   b) Gibt es ein allgemeines Verfahren, das für jedes Programm und jede Eingabe dazu entscheidet, ob die Berechnung terminiert?

4) Schreiben Sie ein Programm, das die Anzahl S(m,n) der Zerlegungen einer natürlichen Zahl m in Summanden, die alle natürliche Zahlen und kleiner gleich n sind, berechnet.

   (Beispiel: S(4,2) = 3, weil 4 = 2+2 = 2+1+1 = 1+1+1+1)

   Hinweis: Leiten Sie zunächst eine geeignete Rekursionsformel her.

5) Gegeben sei die Grammatik G = (Σ,T,Prod,S) mit dem Zeichenvorrat Σ={S,X,Y} T, dem terminalen Zeichenvorrat T={a,c} dem Startsymbol S und der Produktionsmenge Prod = {(S->aXc), (X->XX), (X->AY), (aA->aa), (YA->AY), (Yc->cc)}

   a) Ist die Grammatik kontextfrei?

   b) Geben Sie die von G erzeugte Sprache L(G) an und begründen Sie Ihre Antworten.

6) Welche Datenstrukturen zur Implementierung von Bäumen kennen Sie? Beschreiben Sie diese und eine Prozedur, die das Einfügen eines Knotens in einen Baum leistet.

46112

Grundlagen der Informatik (nicht vertieft)

Herbst 1996

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
	Herbst 1996	46112

Kennzahl: _____

Kennwort: _____

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!

Aufgabe 1

Hexadezimalzahlen sind Zeichenketten über dem Alphabet

$$\mathcal{A} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}.$$

Eine Hexadezimalzahl $h = h_n...h_1h_0$ mit $h_i \in \mathcal{A}$ für $i = 0, ..., n$ stellt die Dezimalzahl

$$h_n * 16^n + h_{n-1} * 16^{n-1} + ... + h_1 * 16 + h_0$$

dar. Dabei hat A den Wert 10, B den Wert 11, C den Wert 12, D den Wert 13, E den Wert 14 und F den Wert 15.

a) Stellen Sie die folgenden Hexadezimalzahlen als Dezimalzahlen dar: 2F, A3E, 100B.
Stellen Sie die folgenden Dezimalzahlen als Hexadezimalzahlen dar: 26, 525, 10000.

Sei $\mathcal{L}$ die Sprache, die aus der Zahl 0 und aus allen Hexadezimalzahlen (beliebiger Länge), die keine führenden Nullen haben, besteht.

b) Beschreiben Sie die Sprache $\mathcal{L}$ entweder durch ein Syntaxdiagramm oder durch eine kontextfreie Grammatik!

c) Geben Sie einen deterministischen Automaten an, der genau die Sprache $\mathcal{L}$ akzeptiert!

Die folgenden Programmieraufgaben können in einer beliebigen höheren Programmiersprache gelöst werden, die die Deklaration von Aufzählungstypen und Feldtypen und die Deklaration von Prozeduren und (rekursiven) Funktionen (bzw. Funktionsprozeduren) erlaubt (etwa Pascal, Modula-2). Geben Sie die von Ihnen verwendete Programmiersprache an!
Als Standardfunktionen für die Arithmetik dürfen im folgenden nur die Operatoren +, -, *, DIV, MOD und die üblichen Vergleichsoperatoren verwendet werden.

d) Es ist ein Datentyp mit Namen *hexzahl* zur Darstellung von Hexadezimalzahlen, die aus 10 Zeichen $h_9...h_1h_0$ bestehen, zu entwickeln. Verwenden Sie dabei einen geeigneten Aufzählungstyp für die Darstellung des Alphabets $\mathcal{A}$!

e) Schreiben Sie eine Prozedur zur Umrechnung von Hexadezimalzahlen in Dezimalzahlen! Die Prozedur soll einen Wertparameter h vom Typ *hexzahl* haben und einen Variablenparameter x, der nach Ablauf der Prozedur den umgerechneten Wert von h enthält.

Fortsetzung nächste Seite!

Aufgabe 2

Gegeben seien die beiden folgenden (kontextfreien) Grammatiken G_1, G_2 mit Variablenmenge (Nichtterminalsymbole) $V = \{S\}$, terminalem Zeichenvorrat $\Sigma = \{a, b\}$, dem Startsymbol S und der Produktionenmenge
$P_1 = \{S \rightarrow SS, S \rightarrow aSb, S \rightarrow \varepsilon\}$ bzw. $P_2 = \{S \rightarrow aSbS, S \rightarrow \varepsilon\}$
(ε = leeres Wort).

a) Zeigen Sie, daß beide Grammatiken dieselbe Sprache L erzeugen!

b) Beschreiben Sie die Sprache L inhaltlich!

c) Zeigen Sie, daß G_1 mehrdeutig ist (d. h. es gibt ein von G_1 erzeugtes Wort mit mindestens zwei verschiedenen Ableitungsbäumen)!

d) Zeigen Sie, daß G_2 eindeutig (=nicht mehrdeutig) ist!

46112

Grundlagen der Informatik (nicht vertieft)

Frühjahr 1997

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr 1997

46112

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: 2

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1

Gegeben sei die Grammatik Γ mit $\{a,b,c\}$ als Menge der Terminalzeichen, den Nichtterminalzeichen Z, A und B, dem Axiom Z und den Produktionsregeln

$Z \rightarrow aA$	$A \rightarrow aA$	$B \rightarrow bB$
$Z \rightarrow bA$	$A \rightarrow bB$	$B \rightarrow c$
	$A \rightarrow c$	

$\mathcal{L}(\Gamma)$ bezeichne den Sprachschatz von Γ.

a) Sei M die Menge aller Zeichenreihen über $\{a,b,c\}$, die mit dem Zeichen c enden. Beweisen oder widerlegen Sie folgende Behauptungen:

   a1) $\mathcal{L}(\Gamma) \subseteq M$.
   a2) $M \subseteq \mathcal{L}(\Gamma)$.

b) Geben Sie einen deterministischen endlichen Automaten an, der genau die Zeichenreihen von $\mathcal{L}(\Gamma)$ akzeptiert.

c) Beschreiben Sie $\mathcal{L}(\Gamma)$ durch einen regulären Ausdruck.

Fortsetzung nächste Seite!

Teilaufgabe 2

a) Von einem Algorithmus A der Art

```
PROCEDURE A(VAR x:ARRAY [1..n] OF REAL)
```

sei bekannt, daß er die Zeitkomplexität $O(n^2)$ hat. Erläutern Sie ausführlich, was diese Aussage bedeutet. In welchem Sinne ist ein Algorithmus mit Zeitkomplexität $O(n)$ "besser" als A ?

b) Beschreiben Sie verbal die Wirkungsweise des Algorithmus QUICKSORT zum Sortieren einer Folge von Zahlen.

c) Welche Zeitkomplexität hat QUICKSORT? Begründen Sie Ihre Antwort.

d) Nennen Sie noch wenigstens zwei andere Sortierverfahren mit einer Zeitkomplexität von gleicher Größenordnung wie die Zeitkomplexität von QUICKSORT.

Teilaufgabe 3

Durch die Funktionsvereinbarung

```
function f(n:nat)nat:
  if n = 0 then 3
  [] n = 1 then 0
  [] n = 2 then 2
  else f(n-2) + f(n-3) endif
```

ist eine Funktion $f: \mathbb{N}_0 \to \mathbb{N}_0$ definiert. Die Funktion $g: \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ sei gegeben durch

$$g(n,x,y,z) = x \cdot f(n+2) + y \cdot f(n+1) + z \cdot f(n).$$

a) Bestimmen Sie (für beliebige $x,y,z \in \mathbb{N}_0$) den Wert von $g(3,x,y,z)$.

b) Beweisen Sie, daß für $n,x,y,z \in \mathbb{N}_0$, $n \geq 1$, gilt: $g(n,x,y,z) = g(n-1,y,x+z,x)$.

c) Geben Sie unter Verwendung von b) einen rekursiven Algorithmus an, der $g(n,x,y,z)$ für beliebige $n,x,y,z \in \mathbb{N}_0$ berechnet.

d) Geben Sie einen iterativen Algorithmus an, der $f(n)$ für beliebiges $n \in \mathbb{N}_0$ berechnet.

Teilaufgabe 4

Gegeben sei die dreistellige Schaltfunktion f mit

$$f(a,b,c) = (a \wedge b) \vee (\neg a \wedge \neg b) \vee (a \wedge \neg c) \vee (\neg a \wedge c) \vee (b \wedge c) \vee (\neg b \wedge \neg c).$$

a) Geben Sie ein Schaltnetz für f an.

b) Beweisen Sie, daß für alle Schaltvariablen a,b,c gilt:

$$f(a,b,c) = (a \vee \neg b \vee c) \wedge (\neg a \vee b \vee \neg c).$$

c) Beweisen oder widerlegen sie folgende Behauptungen:

c1) $f(a,b,c) = f(b,a,c)$ für alle Schaltvariablen a,b,c.
c2) $f(a,b,c) = f(\neg b,\neg a,c)$ für alle Schaltvariablen a,b,c.

46112

Grundlagen der Informatik (nicht vertieft)

Herbst 1997

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Herbst 1997

46112

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!

Verwenden Sie zur Beschreibung von Algorithmen bzw. Datentypen in den Teilaufgaben 1 und 2 eine übliche Programmiersprache (PASCAL, MODULA o.ä.) oder einen entsprechenden "Pseudocode"!

Teilaufgabe 1

Gegeben sei die Grammatik Γ mit der Menge $\Sigma = \{\mathbf{a},\mathbf{b},\mathbf{c}\}$ von Terminalzeichen, den Nichtterminalzeichen Z, D und C, dem Axiom Z und den Produktionsregeln

$Z \to DC$	$D \to C\mathbf{a}$	$C \to \varepsilon$
$Z \to DZ$	$D \to \mathbf{b}$	$C \to \mathbf{c}C$

(ε bezeichne die leere Zeichenreihe.)

$\mathcal{L}(\Gamma)$ bezeichne die von Γ erzeugte Sprache. Für $x \in \Sigma^*$ und $s \in \Sigma$ bezeichne $A_x(s)$ die Anzahl der Vorkommen von s in x. Die Mengen M_1 und M_2 von Zeichenreihen über Σ seien gegeben durch

$$M_1 = \{ x \in \Sigma^* \mid A_x(\mathbf{a}) + A_x(\mathbf{b}) \geq 1 \},$$
$$M_2 = \{ x \in \Sigma^* \mid A_x(\mathbf{a}) + A_x(\mathbf{b}) = A_x(\mathbf{c}) \}.$$

a) Beweisen oder widerlegen Sie folgende Behauptungen:

a1) $\mathbf{cabac} \in \mathcal{L}(\Gamma)$,
a2) $\mathcal{L}(\Gamma) \subseteq M_1$,
a3) $M_1 \subseteq \mathcal{L}(\Gamma)$.

b) Geben Sie einen deterministischen endlichen Automaten an, der genau die Zeichenreihen von M_1 akzeptiert!

c) Gibt es eine reguläre (d.h. Typ-3-) Grammatik, von der genau die Menge M_2 als Sprache erzeugt wird? Begründen Sie Ihre Antwort!

Für das folgende sei $n \in \mathbb{N}_0$ fest vorgegeben. Betrachtet werden nun Zeichenreihen aus Σ^n, d.h. Zeichenreihen der Länge n über Σ. Nehmen Sie an, daß Ihre Programmiersprache zur Realisierung solcher Zeichenreihen einen Datentyp für Reihungen über Σ der Art

array [1..n] **sigma**

(in PASCAL-artiger Notation) mit der üblichen direkten Zugriffsoperation auf die einzelnen Komponenten der Reihung bereitstellt! (**sigma** sei ein Datentyp zur Realisierung von Σ.)

d) Geben Sie einen *iterativen* Algorithmus an, der für eine Zeichenreihe $x \in \Sigma^n$ feststellt, ob $x \in M_1$ ist!

Fortsetzung nächste Seite!

e) Will man die Aufgabe von Teil d) statt durch einen iterativen durch einen *rekursiven* Algorithmus lösen, so ist dies unter den gemachten Voraussetzungen nicht direkt möglich. Als "Umweg" kann man einen allgemeineren Algorithmus der Art

function ALLGLÖSG (x: **array** [1..n] **sigma**, k:**integer**): **boolean**

rekursiv formulieren ("Einbettung"), der einen zusätzlichen Parameter k enthält und für $x = (x_1,...x_n)$ und k mit $1 \le k \le n$ feststellt, ob gilt:

$$A_x(\mathbf{a},k) + A_x(\mathbf{b},k) \ge 1.$$

Dabei bezeichnen $A_x(\mathbf{a},k)$ und $A_x(\mathbf{b},k)$ die Anzahlen der Vorkommen von **a** bzw. **b** unter den ersten k Zeichen $x_1,...,x_k$ von x. Ein Aufruf ALLGLÖSG(x,n) ("Spezialisierung $k = n$") liefert dann offensichtlich eine Lösung der ursprünglichen Aufgabe von Teil d).

Geben Sie einen *rekursiven* Algorithmus für ALLGLÖSG an!

f) Geben Sie einen *iterativen* Algorithmus an, der für eine Zeichenreihe $x \in \Sigma^n$ feststellt, ob $x \in M_2$ ist!

g) Geben Sie analog zu Teil e) eine geeignete Einbettung der Aufgabe von Teil f) in eine allgemeinere Aufgabe an, lösen Sie diese durch einen *rekursiven* Algorithmus, und geben Sie die Spezialisierung an, mit der man daraus eine Lösung der ursprünglichen Aufgabe von Teil f) erhält! (Hinweis: Neben der Verallgemeinerung durch den zusätzlichen Paramater k wie in Teil e) ist eine weitere Verallgemeinerung notwendig.)

Teilaufgabe 2

In einer Wirtschaftsuntersuchung werden n Firmen $F_1,...,F_n$ ($n \ge 2$) und ihre gegenseitigen Geschäftsverbindungen der Art "Firma F_i beliefert Firma F_j mit Waren" ($i,j \in \{1,...,n\}$) betrachtet. Erläutern Sie die Darstellung der Menge aller dieser Beziehungen

a) als Adjazenzmatrix,
b) durch Adjazenzlisten!

Geben Sie jeweils auch die entsprechenden Datentyp-Deklarationen an!

Teilaufgabe 3

Erläutern Sie folgende Adressierungsarten für den Speicher einer Rechenanlage:

a) absolute Adressierung,
b) relative Adressierung,
c) indirekte Adressierung!

Geben Sie jeweils auch eine typische Anwendung an!

46112

Grundlagen der Informatik (nicht vertieft)

Herbst 1998

Prüfungsteilnehmer	Prüfungstermin	Einzel' fungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Herbst

1998

46112

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!

Aufgabe 1

Für reelle Zahlen soll mit Hilfe von geordneten Binärbäumen eine Rechenstruktur aufgebaut werden, die das Einfügen und Löschen erlaubt. Die Programmierung soll in PASCAL erfolgen.

1.1 Geben Sie eine Deklaration für die erforderliche Datenstruktur an!

1.2 Geben Sie Deklarationen für folgende Prozeduren an:

1.2.1 Eine Prozedur KRE mit folgender Spezifikation:
KRE hat den Namen eines Binärbaums als Parameter und erzeugt einen leeren Binärbaum unter diesem Namen.

1.2.2 Eine Prozedur EIN mit folgender Spezifikation:
EIN hat 2 Parameter, und zwar den Namen eines Binärbaums sowie den Wert einer reellen Zahl, und ordnet die reelle Zahl in diesen Binärbaum ein, falls sie noch nicht in ihm enthalten ist.

1.2.3 Eine Prozedur AUS mit folgender Spezifikation:
AUS hat 2 Parameter wie bei **1.2.2** und löscht die reelle Zahl aus dem geordneten Binärbaum, falls sie in ihm enthalten ist.

1.3 Skizzieren Sie den geordneten Binärbaum mit dem Namen x, der entsteht, wenn ein Programmlauf folgende Prozeduraufrufe in der angegebenen Reihenfolge ausführt:

KRE(x); EIN(x,3.1); EIN(x,5.7); EIN(x,3.1);
EIN(x,5.6); EIN(x,2.4); AUS(x,3.1); EIN(x,7.0); EIN(x,4.8);
EIN(x,3.0); EIN(x,3.2); EIN(x,1.7); EIN(x,2.6); AUS(x,4.8);

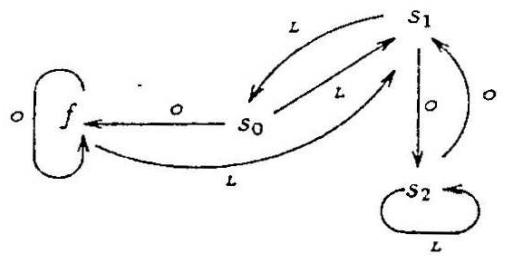
Aufgabe 2

Gegeben sei das binäre Alphabet $B = (O, L)$. Die Menge $P \subset B^*$ sei die Menge aller durch 5 teilbaren und $H \subset B^*$ die Menge aller durch 6 teilbaren Binärzahlen. Die leere Zeichenreihe ε werde nicht als Binärzahl angesehen, ist also weder in P noch in H enthalten.

2.1 Konstruieren Sie einen endlichen Automaten A_P, der genau die Binärzahlen aus P akzeptiert!

Hinweis: Ein Automat A_P kann z.B. mit Hilfe des gewöhnlichen Divisionsalgorithmus für Binärzahlen gefunden werden.

2.2 Der folgende Graph stellt einen endlichen Automaten A_H über B dar, der mit s_0 als Anfangs– und f als Endzustand genau die Binärzahlen aus H akzeptiert. Konstruieren Sie mit Hilfe dieses Automaten A_H die Menge H als reguläre Menge!



Fortsetzung nächste Seite!

Aufgabe 3

Gegeben sei ein endlicher ungerichteter Graph G mit der Knotenmenge $V = (v_1, v_2, \ldots, v_n)$, $n \geq 1$. G enthalte weder Schleifen noch Mehrfachkanten. Zur Darstellung des Graphen werde die Adjazenzmatrix

$$A = (a_{i,j})_{i,j=1,2,\ldots,n}$$

verwendet. Nach den oben genannten Festlegungen gilt:

- $a_{i,j} = \begin{cases} 0, & \text{falls } v_i \text{ und } v_j \text{ nicht durch eine Kante verbunden sind} \\ 1 & \text{sonst} \end{cases}$
- $a_{i,j} = a_{j,i}$ für $i, j = 1, 2, \ldots n$
- $a_{i,i} = 0$ für $i = 1, 2, \ldots n$

Seien u und v zwei Knoten aus V. Dann heißt eine endliche Knotenfolge $w = (w_0, w_1, w_2, \ldots, w_r)$ aus V ein Weg der Länge r zwischen u und v genau dann, wenn w folgenden Bedingungen genügt:

- $r \geq 0$
- Sei $p(\varrho)$ der eindeutige Index mit $v_{p(\varrho)} = w_\varrho$ für $\varrho = 0, 1, 2, \ldots r$. Dann gilt $a_{p(\varrho),p(\varrho+1)} = 1$ für $\varrho = 0, 1, 2, \ldots, r-1$.
- $((w_0 = u) \wedge (w_r = v)) \vee ((w_0 = v) \wedge (w_r = u))$

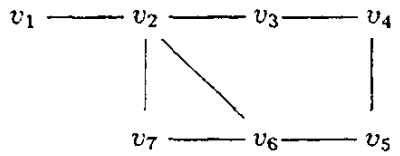
Als Matrix der kürzesten Weglängen von G wird die Matrix

$$W = (w_{i,j})_{i,j=1,2,\ldots,n}$$

bezeichnet, deren Komponenten folgendermaßen definiert sind:

$$w_{i,j} := \begin{cases} \ell, & \text{falls es einen Weg zwischen } v_i \text{ und } v_j \text{ gibt und} \\ & \ell \text{ die Länge eines kürzesten Weges zwischen } v_i \text{ und } v_j \text{ ist} \\ 0, & \text{falls es keinen Weg zwischen } v_i \text{ und } v_j \text{ gibt} \end{cases}$$

3.1 Geben Sie W für den folgenden Graphen an!



3.2 Nennen Sie Knotenpaare (v_i, v_j), für die $w_{i,j}$ stets 0 ist, und begründen Sie Ihre Antwort!

3.3 Schreiben Sie eine Prozedur `WEGE` in PASCAL mit folgender Spezifikation:

`WEGE` hat 2 Eingabe– und 1 Ausgabeparameter. Die Eingabeparameter sind der Wert von n und der Name von A. Der Ausgabeparameter ist der Name von W. Die Prozedur berechnet die Matrix W aus n und A.

Hinweis: Es ist empfehlenswert, das Programm für die Prozedur `WEGE` mit Hilfe des Warshall–Algorithmus zu entwickeln.

46112

Grundlagen der Informatik (nicht vertieft)

Frühjahr 1999

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr

1999

46112

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:

$G = (\{E\}, \{ a, b, +, *, (,) \}, P, E)$
mit
$P = \{ E \rightarrow E+E \mid E*E \mid (E) \mid a \mid b \}$.

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 $\mathcal{A} = (S, \Sigma, \delta, s0, S1)$ ein deterministischer Automat mit Zustandsmenge $S = \{p, q, r, s0, s1\}$ 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 Σ.

iii) Einen Datentyp "TYPE Transition =..." zur Repräsentation von Zustandsübergangsfunktionen (d.h. δ: S x $\Sigma \rightarrow$ S soll als ein Element von "Transition" dargestellt werden können).

b) Es sei angenommen, dass S1 aus genau einem Endzustand s1 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 s0 = p, den Endzustand s1 = r und folgende Übergangsfunktion δ:

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

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.

46112

Grundlagen der Informatik (nicht vertieft)

Herbst 1999

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Herbst

1999

46112

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: 2

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1:

1. Skizzieren Sie die Struktur des klassischen Universalrechners (URA) nach von Neumann!
2. Beschreiben Sie den Befehlsablauf des Befehls „*Speichere Akkuinhalt in Speicherzelle 200*“!
3. Beschreiben Sie den Befehlsablauf des Befehls „*Springe nach Befehl in Speicherzelle 100*“!

Teilaufgabe 2:

Erklären Sie den Unterschied zwischen horizontaler und vertikaler Mikroprogrammierung!

Fortsetzung nächste Seite!

Teilaufgabe 3:

Gegeben sind folgende Ableitungsregeln für eine Sprache in Backus-Naur-Form:

S ::= aSb | A

A ::= cAc | c

Dabei sind **a, b, c** terminale Symbole und **S, A** nichtterminale Symbole.

1. Welche Symbolfolgen, die nur Terminale enthalten, sind aus **S** ableitbar?

2. Welche der Symbolfolgen $\mathbf{a^2\,b^2}$, **acb**, $\mathbf{a^3cb}$ und **c** sind gültige Wörter der gegebenen Sprache? Begründen Sie, warum die verbleibenden Symbolfolgen keine gültigen Wörter der Sprache sind! Geben Sie für die gültigen Wörter die entsprechenden Ableitungsbäume an!

Teilaufgabe 4:

1. Definieren Sie den Begriff des Gültigkeitsbereichs in blockstrukturierten Programmiersprachen!

2. Definieren Sie einen abstrakten Datentyp *„Symboltabelle"*!
   Die Aufgabe einer Symboltabelle besteht darin, den in einem Programm verwendeten Bezeichnern Informationen zuzuordnen. Insbesondere soll die Gültigkeit eines Bezeichners ableitbar sein. Der abstrakte Datentyp soll folgende Funktionen zur Verfügung stellen:

   - Erzeugen einer Symboltabelle
   - Funktion zum Eintreten in einen neuen Block
   - Funktion zum Eintragen eines Bezeichners und der zugehörigen Information in die Symboltabelle
   - Funktion zum Verlassen eines Blocks
   - Test, ob ein Bezeichner im aktuellen Block enthalten ist
   - Funktion, die zu einem Bezeichner die zugehörige Information liefert

Teilaufgabe 5:

1. Erklären Sie folgende Begriffe:
   - Seite
   - Kachel

   Wie ist eine virtuelle Adresse aufgebaut (nur reines Paging)?

2. Skizzieren Sie den Abbildungsmechanismus von virtuellen auf physikalische Adressen!

3. Welcher Hardwarevorgang (nur ein Begriff) und welche Softwareroutinen (hier etwas genauer) laufen ab, wenn eine Seite angesprochen wird, die nicht im Arbeitsspeicher ist?

4. Beschreiben Sie die Ersetzungsstrategie LRU.