Sammlung aller Staatsexamensaufgaben der Prüfungsnummer

46113

Theoretische Informatik (nicht vertieft)

46113

Theoretische Informatik (nicht vertieft)

Herbst 1998

Prüfungsteilnehmer	Prüfungstermin	Einz üfungsnummer
Kennzahl: _____	**Herbst**	**46113**
Kennwort: _____	**1998**	
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 3

Bitte wenden!

Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1:

Gegeben sei die Grammatik G mit {a,b} als Menge der Terminalzeichen, den Variablen S,A,B, der Startvariablen S und den Produktionen

$S \to abA$, $S \to bB$, $S \to a$, $A \to B$, $A \to a$, $B \to aS$, $B \to b$.

Sei L die von G erzeugte Sprache.

a. Geben Sie eine zu G äquivalente Typ-3-Grammatik in Normalform an!

b. Konstruieren Sie einen nicht-deterministischen Automaten, der L akzeptiert!

c. Konstruieren Sie einen deterministischen Automaten, der L akzeptiert!

d. Geben Sie einen regulären Ausdruck für L an!

Aufgabe 2:

Sei $L = \{ w \mid l_a(w) = l_b(w) \}$ die Menge aller Wörter über {a,b}, die gleich viele Vorkommen von a und b haben ($l_x(w)$ = Anzahl der Vorkommen von x in w).

a. Zeigen Sie, dass L eine kontextfreie Sprache ist, durch Konstruktion eines Kellerautomaten, der L akzeptiert oder durch Konstruktion einer L erzeugenden kontextfreien Grammatik!

b. Ist L regulär? (Begründung)

Aufgabe 3:

Sei die (partielle) Funktion $f : N^2 \to N$ (N Menge der natürlichen Zahlen einschließlich 0) gegeben durch $f(x,y)$ = **if** $x > y$ **then** $f(x - y, y)$ **else if** $x < y$ **then** $f(x, y - x)$ **else** x

a. Berechnen Sie f(18,12) !

b. Bestimmen Sie den Definitionsbereich Df von f!

c. Zeigen Sie, dass f(x,y) für $(x,y) \in Df$ der größte gemeinsame Teiler von x und y ist!

d. Ist f primitiv rekursiv? (Begründung!)

e. Ist $f' = \langle (x,y) \to$ **if** $(x,y) \in Df$ **then** $f(x,y)$ **else** $0 \rangle$ primitiv rekursiv? (Begründung!)

f. Geben Sie ein GOTO-Programm für f an!(Verwendung von Makros der Form „$x_i := x_j \dot{-} x_k$" zur Berechnung der modifizierten Differenz ist erlaubt.)

Thema Nr. 2

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1:

Ist folgendes Problem entscheidbar? (Begründung!)

- gegeben: Turingmaschine M
- entscheide: M berechnet die Identitätsfunktion $< x \rightarrow x >$ (auf **N**)

Aufgabe 2:

Sei F die Formel $((A \rightarrow B) \wedge (B \rightarrow C) \wedge (C \rightarrow A)) \rightarrow (C \rightarrow B)$

a. Ist F erfüllbar? (Begründung!)

b. Ist F eine Tautologie? (Begründung!)

Aufgabe 3:

Sei F die Formel $\forall z \forall x \exists y P(f(x, y), z)$.

Seien N und Z zwei Strukturen mit den Trägermengen U^N = Menge der natürlichen Zahlen (einschließlich 0) bzw. U^Z = Menge der ganzen Zahlen, P^N bzw. P^Z die jeweilige Identitätsrelation und f^N bzw f^Z die jeweilige Additionsfunktion.

a. Ist N Modell von F? (Begründung!)

b. Ist Z Modell von F? (Begründung!)

c. Ist F erfüllbar? (Begründung!)

d. Ist F allgemeingültig? (Begründung')

e. Hat F pränexe Normalform? (Begründung!)

f. Geben Sie zu F eine äquivalente pränexe Normalform F´ an!

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2000

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr**	**46113**
Kennwort: _____	**2000**	
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 3

Bitte wenden!

Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1:

Gegeben sei die Grammatik Γ mit der Menge $\{a,b,c\}$ von Terminalzeichen, der Menge $\{S,A,B,C\}$ von Nicht-Terminalzeichen, dem Startsymbol S und den Produktionsregeln

$$S \to \varepsilon \mid aA \mid bB \mid cb$$
$$A \to \varepsilon \mid aC \mid bB \mid cB \mid ccb$$
$$B \to \varepsilon \mid bB$$
$$C \to \varepsilon \mid aC \mid bB \mid cb$$

(ε bezeichne das leere Wort.) $\mathcal{L}(\Gamma)$ sei die von Γ erzeugte Sprache.

a) Beweisen Sie:

a1) $aaabb \in \mathcal{L}(\Gamma)$.

a2) c kommt in jedem Wort von $\mathcal{L}(\Gamma)$ höchstens zweimal vor.

a3) Ist $w \in \mathcal{L}(\Gamma)$ ein Wort, das genau ein c enthält, so enthält w genau ein a oder genau ein b.

b) Geben Sie einen nicht-deterministischen endlichen Automaten an, der genau die Elemente von $\mathcal{L}(\Gamma)$ akzeptiert!

Teilaufgabe 2:

Sei $L = \{a^n b c^{2n} \mid n \in \mathbb{N}_0\}$ eine Sprache über dem Alphabet $\{a,b,c\}$. Beweisen Sie:

a) L ist nicht regulär.

b) L ist kontext-frei.

Teilaufgabe 3:

Die Funktionen $f: \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}_0$ und $g: \mathbb{N}_0 \to \mathbb{N}_0$ seien definiert durch:

$$f(n,m) = \begin{cases} n-m, & \text{falls } n \geq m \\ m-n & \text{sonst,} \end{cases} \qquad \text{und} \qquad g(n) = \begin{cases} \sqrt{n}, & \text{falls } \sqrt{n} \in \mathbb{N}_0 \\ \text{undefiniert} & \text{sonst.} \end{cases}$$

Beweisen Sie:

a) f ist primitiv-rekursiv.

b) g ist partiell-rekursiv.

Hinweis: Sie dürfen ohne Beweis voraussetzen, dass die arithmetischen Grundfunktionen Addition, Subtraktion und Multiplikation primitiv-rekursiv sind!

Thema Nr. 2

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1:

a) Gegeben ist eine formale Grammatik $\mathcal{G} = (S, \Sigma, P, s_0)$. Welche Bedingungen müssen die Produktionen P erfüllen, damit die Grammatik

i) rechtslinear, ii) kontextfrei, iii) kontextsensitiv ist?

Die von $\mathcal{G}$ erzeugte Sprache wird mit $L(\mathcal{G})$ bezeichnet. $L(\mathcal{G})$ heißt "vom Typ 3", wenn $\mathcal{G}$ rechtslinear ist, "vom Typ 2", wenn $\mathcal{G}$ kontextfrei ist und "vom Typ 1", wenn $\mathcal{G}$ kontextsensitiv ist.

b) Charakterisieren Sie Sprachen vom Typ 3 und vom Typ 2 mit Hilfe von Automatentypen!

c) Gegeben sei die Grammatik $\mathcal{G} = (\{A, B\}, \{0, 1\}, P, A)$ mit
$P = \{ A \to 0B, B \to 01B, B \to 10B, B \to 1 \}$.

i) Geben Sie einen regulären Ausdruck mit Sprache $L(\mathcal{G})$ an!
ii) Konstruieren Sie einen nichtdeterministischen endlichen Automaten, der die Sprache $L(\mathcal{G})$ akzeptiert!
iii) Konstruieren Sie einen minimalen deterministischen endlichen Automaten, der die Sprache $L(\mathcal{G})$ akzeptiert!

d) Geben Sie eine Typ 2-Sprache $\mathcal{L}2$ an, die nicht vom Typ 3 ist, zusammen mit einer kontextfreien Grammatik, die $\mathcal{L}2$ erzeugt!

e) Geben Sie eine Typ 1-Sprache $\mathcal{L}1$ an, die nicht vom Typ 2 ist, zusammen mit einer kontextfreien Grammatik, die $\mathcal{L}1$ erzeugt!

Teilaufgabe 2:

Gegeben sei die folgende Funktionsdefinition:

```
function f (n: Nat) : Nat;
if n = 0 then 0 else n+f(n-1) end;
```

a) Wie lautet das zur obigen Definition gehörige Funktional $\Phi: [N^\perp \to N^\perp] \to [N^\perp \to N^\perp]$? Zeigen Sie, dass die Funktion

$$g: N^\perp \to N^\perp \quad \text{mit } g(n) = n*(n+1)/2 \text{ falls } n \neq \perp,\ g(\perp) = \perp,$$

ein Fixpunkt von Φ ist! (Hierbei bezeichnet $N^\perp$ die Menge der natürlichen Zahlen erweitert um das Element $\perp$.)

b) Beweisen Sie durch Induktion, dass für alle natürlichen Zahlen n gilt: $f(n) = n*(n+1)/2$.

46113

Theoretische Informatik (nicht vertieft)

Herbst 2000

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst**	**46113**
Kennwort: _____	**2000**	
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

Bitte wenden!

Thema Nr. 1

Aufgabe 1

Zur Darstellung einer Gleitkommazahl arbeiten wir mit dem Alphabet S = {´+´, ´–´, ´.´, ´E´, ´0´, ´1´, ´2´, ´3´, ´4´, ´5´, ´6´, ´7´, ´8´, ´9´}. Daraus bilden wir Zeichenketten A, .. ,G $\in S^*$, die wir schließlich mit Hilfe der Verkettungsoperation ´o´ zu Gleitkommazahlen Z kombinieren: Z = A o B o C o D o E o F o G. Für die Zeichenfolgen A, .. , G gelten dabei die folgenden Bedingungen:

Zeichenreihe	*Länge l*	darf folgende Zeichen enthalten	Einschränkungen
A	$0 \le l_A \le 1$	´+´, ´–´	
B	$l_B \ge 1$	´0´, ´1´, ´2´, ´3´, ´4´, ´5´, ´6´, ´7´, ´8´, ´9´	
C	$0 \le l_C \le 1$	´.´	$l_C = 1 \Leftrightarrow l_D \ge 1$
D	$l_D \ge 0$	´0´, ´1´, ´2´, ´3´, ´4´, ´5´, ´6´, ´7´, ´8´, ´9´	
E	$0 \le l_E \le 1$	´E´	$l_E = 1 \Leftrightarrow l_G \ge 1$
F	$0 \le l_F \le 1$	´+´, ´–´	$l_F = 1 \Rightarrow l_G \ge 1$
G	$l_G \ge 0$	´0´, ´1´, ´2´, ´3´, ´4´, ´5´, ´6´, ´7´, ´8´, ´9´	

Zulässige Gleitkommazahlen wären damit etwa „+5“, „9.123“, „1298.222E+2“, „-2E2“. Nicht zugelassen sind dagegen z.B. „.12“, „E12“ oder „1E+“. Verwenden Sie in den folgenden Aufgaben als Abkürzung das Symbol t mit t $\in$ {´0´, .. , ´9´}!

a) $L_Z \subset S^*$ sei die Sprache, die man durch Bildung aller möglichen Gleitkommazahlen gemäß der in obiger Tabelle angegebenen Regeln erhält.
Zeigen Sie, dass L_Z vom Typ Chomsky-3 ist, indem Sie eine erzeugende rechtslineare Grammatik G dafür angeben!

b) Zeichnen Sie das Zustands-Übergangsdiagramm eines endlichen Automaten, der L_Z akzeptiert!

c) Geben Sie einen regulären Ausdruck für L_Z an!

Fortsetzung nächste Seite!

Aufgabe 2

Wir betrachten folgendes Programmfragment P (in Pascal-ähnlicher Notation):

```
i := 1; f := 1;
while i ≤ n do
        begin
        f := f*i;
        i := i+1;
        end;
```

a) Welche Funktion f wird durch dieses Programmfragment berechnet?

b) Bezüglich der Berechenbarkeit von Funktionen durch imperative Programme unterscheidet man zwischen FOR- (oder LOOP-) und WHILE-Berechenbarkeit.
Charakterisieren Sie kurz diese beiden Berechenbarkeitsbegriffe und ihre Beziehung zueinander!

c) Zu welcher der beiden in b) genannten Berechenbarkeitsklassen gehört die Funktion f aus Teilaufgabe a)?
Begründen Sie Ihre Antwort!

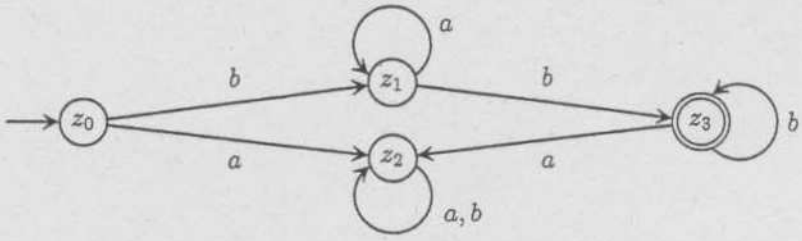
Aufgabe 3

a) Geben Sie eine Turing-Maschine TM zur Berechnung der Vorgängerfunktion *pred* einer natürlichen Zahl $n \geq 1$ an! Dabei soll die Zahl n in Binärform (höchstwertiges Bit links) ohne führende Nullen auf dem Band von TM dargestellt sein und der Schreib-Lesekopf anfangs auf dem höchstwertigen Bit stehen. Nach der Berechnung dürfen führende Nullen auftreten. Außerhalb der Darstellung von n steht an allen anderen Stellen des Bandes das Leerzeichen ´#´.

b) Wenden Sie Ihre Turingmaschine zur Berechnung von *pred(24)* an!
Geben Sie dabei für alle Zwischenzustände jeweils die Bandbelegung und den Zustand von TM an!

Thema Nr. 2

Aufgabe 1



Welche der folgenden regulären Ausdrücke α_i beschreiben genau die Sprache, die der endliche Automat mit obigem Zustandsgraph akzeptiert? (Es kann mehrere solche Ausdrücke geben.)

α_1 = b(a)*b
α_2 = b(a)*b(b)*
α_3 = (b(a)*b(b)* | a(a| b)*a(b)*)
α_4 = b(a)*(b)*
α_5 = b(a)*(b)*b

Aufgabe 2

2.1 „Eine Sprache L ist genau dann regulär, wenn *Bedingung*."
Geben Sie vier verschiedene Bedingungen an, für die diese Aussage korrekt ist!

2.2 Gibt es drei Sprachen L_1, L_2, L_3 mit $L_1 \subseteq L_2 \subseteq L_3$,
so dass L_1 und L_3 vom Typ 2 (kontextfrei) sind, aber L_2 nicht?

Geben Sie entweder drei solche Sprachen an oder einen Beweis, dass dies nicht möglich ist!

Fortsetzung nächste Seite!

Aufgabe 3

Ein Palindrom ist ein Wort, das vorwärts und rückwärts gelesen gleich ist, z.B. *anna* oder *uhu*. Das leere Wort ε ist auch ein Palindrom.

Sei $\Sigma = \{a, b\}$ und sei L die Menge aller Wörter über Σ, die Palindrome sind.

3.1 Geben Sie eine Grammatik $G = (V, \Sigma, P, S)$ an mit $L = L(G)$!

3.2 Beweisen Sie, dass L nicht regulär ist!

3.3 Geben Sie eine deterministische Turingmaschine $M = (Z, \Sigma, \Gamma, \delta, s, _, \{e\})$ an, die genau die Palindrome über Σ akzeptiert, für die also $T(M) = L$ ist!

Dabei ist Z die Zustandsmenge, $\Sigma = \{a, b\}$ das Eingabealphabet, $\Gamma = \{a, b, _\}$ das Bandalphabet, _ das Leerzeichen, $s \in Z$ der Anfangszustand, $e \in Z$ der Endzustand (hier reicht ein einziger) und δ die Überführungsfunktion.

Geben Sie zu jedem von Ihnen verwendeten Zustand in Z eine informelle Erläuterung, welchem Zweck der Zustand dienen soll, etwa in folgender Form: „Zustand r_a: nach rechts laufen bis Wortende, anschließend prüfen, ob Wort mit a endet"!

3.4 Geben Sie die Folge der Konfigurationen an, die M jeweils durchläuft für das Eingabewort *aba*, für das Eingabewort *baab* und für das Eingabewort *abaa*!

Aufgabe 4

Sei Σ ein Alphabet.

4.1 Geben Sie die Definition dafür an, dass eine Menge $L \subseteq \Sigma^*$ rekursiv aufzählbar ist!

4.2 Geben Sie die Definition dafür an, dass eine Menge $L \subseteq \Sigma^*$ semi-entscheidbar ist!

4.3 Wie hängen die in den Teilaufgaben 4.1 und 4.2 definierten Eigenschaften zusammen?

4.4 Beweisen Sie: Sind L_1 und L_2 rekursiv aufzählbar, dann ist auch $L_1 \cup L_2$ rekursiv aufzählbar!

4.5 Beweisen Sie: Sind L_1 und L_2 rekursiv aufzählbar, dann ist auch $L_1 \cap L_2$ rekursiv aufzählbar!

Hinweis: Benutzen Sie den Zusammenhang zwischen rekursiv aufzählbar und semi-entscheidbar!

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2001

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr**	
Kennwort: _____	**2001**	**46113**
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

Bitte wenden!

Thema Nr. 1

1. a) Man gebe einen deterministischen Kellerautomaten an, der $L = \{a^k b^{2k} a \mid k \geq 1\}$ akzeptiert.

   # sei das Kellerbodenzeichen, als weitere Kellerzeichen verwende man A, B, ...,
   die Zustände bezeichne man mit z_0, z_1, ...; z_0 sei der Anfangszustand.

   Die Überführungsfunktion schreibe man so:

   (Zustand z_i, Eingabezeichen a oder b, oberstes Kellerzeichen) $\rightarrow$
   (Zustand z_j, das das oberste Kellerzeichen ersetzende Wort).

   b) Man gebe eine kontextfreie Grammatik für L an.

   c) Fragen:

   α) Ist auch der $L_1 = \{a^k b^{2k} a^m \mid k, m \geq 1\}$ deterministisch kontextfrei?

   β) Ist L_1 kontextfrei?

   Man gebe jeweils eine Begründung an.

2. Sei $L_1 = \{a^n b^m c^k \mid n, m, k \geq 0, m = 0 \text{ oder } k = 0\}$
   und $L_2 = \{a^n b^m c^k \mid n, m, k \geq 0, m = n \text{ oder } k = n\}$.

   a) Welche der zwei Sprachen ist regulär, welche nicht?

   b) Sind beide kontextfrei?

   Hinweis: Um zu zeigen, dass eine der Sprachen regulär ist, genügt es, einen regulären Ausdruck oder einen endlichen Automaten für die Sprachen anzugeben.

   Zum Nachweis der Nichtregularität können Sie statt des Pumping Lemmas auch Operationen auf regulären Sprachen verwenden, sowie die Tatsache, dass $L_3 = \{a^n b^n \mid n > 0\}$ nicht regulär ist.

3. Man betrachte die Grammatik G mit den Produktionen

   $S \rightarrow AS \mid AB \mid SS$
   $B \rightarrow CC$
   $A \rightarrow a$
   $C \rightarrow c$

   a) Man vereinfache die Grammatik durch Verringerung der Zahl der Variablen (A, B, C).

   b) Man zeige, dass die erzeugte Sprache $L(G)$ regulär ist — etwa durch Angabe eines regulären Ausdrucks oder eines akzeptierenden endlichen Automaten (mit Begründung).

Fortsetzung nächste Seite!

4. Man betrachte die allgemeine Chomsky-Grammatik G (Grammatik vom Typ 0) mit folgenden 6 Produktionen:

(1) $S \rightarrow LaR$
(2) $La \rightarrow LB$
(3) $Ba \rightarrow aaB$
(4) $BR \rightarrow aaR$
(5) $L \rightarrow \varepsilon$
(6) $R \quad \varepsilon$

a) Welche Sprache (Teilmenge von $\{a\}^*$) erzeugt G?

Hinweis: Man betrachte zunächst nur die ersten 4 Produktionen (ohne die Löschproduktion für L und R) und gebe die Satzformen an, die kein B mehr enthalten und mit den obigen 4 Regeln erzeugt werden können.

b) Ist diese Sprache kontextfrei?

Thema Nr. 2

1. Beantworten Sie die folgende Aussage mit ja oder nein! (kurze Begründung)

   $(ab \mid a)^* a$ ist mit $a (ba \mid a)^*$ äquivalent

2. Zeigen Sie mit dem Pumping Lemma, dass die folgende Sprache L nicht regulär ist!

   $L = \{a^n b a^m b a^{n+m} : n, m \geq 1\}$

3. Sei L die Sprache

   $\{w \in \{0,1\}^* :$ der Teilstring 011 ist nicht in w enthalten$\}$

   a) Geben Sie das Zustandsdiagramm für einen endlichen Automaten an, der L akzeptiert!

   b) Geben Sie eine reguläre Grammatik G an, die L erzeugt!

4. Sei $G = (V,\Sigma,R,S)$ eine kontextfreie Grammatik, wobei $V = \{a,b,A,S\}$, $\Sigma = \{a,b\}$ und die Menge R der Regeln oder Produktionen wie folgt gegeben ist:

   $S \quad aAS \mid a$

   $A \quad SbA \mid SS \mid ba$

   Geben Sie für das Wort

   aabbaa

   einen Parsebaum an!

5. Betrachten Sie die durch folgende Produktionen definierte kontextfreie Grammatik G mit den Regeln:

   $S \quad aAB$

   $A \quad aBB$

   $A \quad a$

   $B \quad bCC$

   $B \quad b$

   $C \quad c.$

   Geben Sie für die Sprache

   $L(G) = \{w \in \{a,b,c\}^* : S \Rightarrow^*_G w\}$

   einen regulären Ausdruck an!

   Hinweis: L(G) sei die Menge aller Wörter, die nur aus Terminalsymbolen bestehen und die von der Grammatik G abgeleitet werden können.

Fortsetzung nächste Seite!

6. Ist die folgende Funktion $f : \{0,1,\ldots,9\}^* \quad \{0,1\}$ rekursiv?
   Die Eingabe x wird als Dezimialzahl interpretiert. Es ist $f(x) = 1$ genau dann, wenn es in der Dezimaldarstellung von $\prod$ einen geschlossenen Block von mindestens x Siebenen gibt. Beweisen Sie Ihre Antwort.

7. Eine Turingmaschine (Einbandturingmaschine)

$$M = (Q,\Sigma,\Gamma,\delta,s,h,\#)$$

wird durch

(i) eine endliche Zustandsmenge Q

(ii) ein endliches Eingabealphabet Σ

(iii) ein endliches Arbeitsalphabet Γ, wo $\Sigma \subseteq \Gamma$

(iv) eine Übergangsfunktion

$$\delta: (Q - \{h\}) \times \Gamma \quad Q \times \Gamma \times \{L,R,N\}$$

(v) einen Startzustand s in Q

(vi) ein Leerzeichen # in $(\Gamma - \Sigma)$

beschrieben.

Sie startet bei der ersten Zelle des Bandes, die ein Leerzeichen enthält, und stoppt, wenn das nächste Leerzeichen gelesen wird.

a) Beschreiben Sie anschaulich eine Turingmaschine M, die die Nachfolgerfunktion $S(x) = x + 1$ berechnet, bei der die Eingabe und Ausgabe in der Unärdarstellung gegeben werden!

b) Geben Sie die Übergangsfunktion δ für diese Turingmaschine an!

46113

Theoretische Informatik (nicht vertieft)

Herbst 2001

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Herbst 2001

46113

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 2

Thema Nr. 1

1. (Automatentheorie) Für $n \geq 0$ sei $L_n =_{\text{def}} \{a, b\}^* a \{a, b\}^n a \{a, b\}^*$ die Menge aller Wörter aus $\{a, b\}^*$, in denen der Buchstabe a an zwei Stellen mit genau n dazwischenliegenden Buchstaben vorkommt. Geben Sie einen nichtdeterministischen endlichen Automaten mit $n + 3$ Zuständen an, der L_n akzeptiert (erkennt)!

2. (Formale Sprachen) Geben Sie eine kontextfreie Grammatik an, die die Sprache $L =_{\text{def}} \{a^n b^m : 0 \leq n \leq m\}$ erzeugt! Zeigen Sie weiter, dass diese Sprache nicht regulär ist!

3. (Berechenbarkeit) Zeigen Sie, dass die Sprache $\{(13n^3 - 17m)^2 : n, m \in \mathbb{N}\}$ rekursiv aufzählbar ist!

4. (Komplexität) Es sei $\exp : \mathbb{N}^2 \to \mathbb{N}$ mit $\exp(x,y) =_{\text{def}} x^y$ die Exponentialfunktion.

   a) Ist exp in polynomieller Zeit berechenbar?

   b) Zeigen Sie durch Angabe eines geeigneten Algorithmus, dass die Menge $\{(x,y,z) : x^y = z\}$ in polynomieller Zeit entscheidbar ist!

Alle Antworten sind zu begründen.

Thema Nr. 2

Aufgabe 1 (Automatentheorie)

Gegeben seien das Alphabet $I = (A,B,C)$ und die regulären Mengen R_1 und R_2 über dem Alphabet I:

$$R_1 = \{\{AB\}^*\{C\}\{C\}^*\}^* \text{ und } R_2 = \{\{AB\}^*\{C\}\}^*.$$

Beweisen Sie, dass $R_1 = R_2$ ist!

Hinweis: Für den Beweis empfiehlt es sich, endliche Automaten A_1 und A_2 zu konstruieren, deren akzeptierte Wortmengen $L(A_1) = R_1$ und $L(A_2) = R_2$ sind.

Aufgabe 2 (Formale Sprachen)

Gegeben seien das terminale Vokabular $V_T = (a,b,c)$ und die drei Wortmengen L_1, L_2, L_3, über dem Vokabular V_T:

$$L_1 = \{a^{2n}b^nc^n\}_{n>0}, \quad L_2 = \{a^nb^{n+m}c^m\}_{n,m>0}, \quad L_3 = \{a^{n+m}b^nc^{n+m}\}_{n,m>0}.$$

1. Diese drei Wortmengen sind *kontextabhängige* Sprachen. Eine von ihnen ist darüber hinaus *kontextfrei*. Stellen Sie fest, welche das ist, und konstruieren Sie dafür eine CHOMSKY-2-Grammatik!

2. Beweisen Sie für **eine** der beiden anderen Sprachen, die Sie selbst auswählen können, dass sie **nicht** *kontextfrei* ist, und konstruieren Sie dafür eine CHOMSKY-1-Grammatik!

Aufgabe 3 (Berechenbarkeit)

Gegeben sei die Funktion c: $\mathbb{N}_0 \to \mathbb{N}_0$ mit $c(x) = \text{entier}(\sqrt[3]{x})$ [1].

1. Beweisen Sie, dass die Funktion $c(x)$ primitiv rekursiv ist!

Hinweis: Für den Beweis können Sie anstelle des *Modells der primitiv rekursiven Funktionen* ein dazu äquivalentes *Programmiermodell* verwenden.

2. Geben Sie eine vollständige μ-rekursive Herleitung der Funktion $c(x)$ an, in der eine Minimalisierung verwendet wird!

[1] *entier*(a) ist die größte ganze Zahl, die kleiner oder gleich a ist.

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2002

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr**	**46113**
Kennwort: _____	**2002**	
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (nicht vertieft studiert)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 4

Bitte wenden!

Thema Nr. 1

Teilaufgabe 1

Für einen regulären Ausdruck R sei L(R) die durch R dargestellte Wortmenge.

Im Folgenden sind jeweils Paare von Wortmengen über dem Alphabet {a,b} gegeben, die durch reguläre Ausdrücke sowie die Mengenoperationen $\cap$ und – beschrieben sind.
(Es ist $W - V = \{u \in W \mid u \notin V\}$; das + in den regulären Ausdrücken wird in der Literatur manchmal auch als | geschrieben.)

Man gebe jeweils an - mit Begründung -, ob die beiden Mengen gleich sind oder nicht und ob eine in der anderen erhalten ist.

a) L (((ab)*(ba)*)*) und L ((abba)*)

b) {a, b}* – L ((aaa(a + b)*) und L ((b + ab + aab + aaaa) (a+b)* + ε)

c) L (a(a+b)*) $\cap$ L ((a+b)*b) und L (a(a+b)*b)

Teilaufgabe 2

Sei k eine feste natürliche Zahl ($k \neq 0$) und {a,b} ein Alphabet.

Welche der folgenden Sprachen ist regulär, kontextfrei oder kontextsensitiv?

Man gebe jeweils eine Grammatik oder einen akzeptierenden Automaten für die Sprache an - mit Begründung - sowie einen Beweis dafür, dass die Sprache nicht regulär bzw. nicht kontextfrei ist!

a) $\{a^k b^m a^k b^m \mid m \geq 1\}$

b) $\{a^k a^k b^m b^m \mid m \geq 1\}$

c) $\{a^n b^m a^n b^m \mid n, m \geq 1\}$

Fortsetzung nächste Seite!

Teilaufgabe 3

Man untersuche die folgende rekursiv definierte Funktion:

$$f(n) = \begin{cases} n & \text{für } n \leq 7 \\ f(f(n \text{ div } 8) + n \bmod 8) & \text{sonst} \end{cases}$$

Dabei sei div die ganzzahlige Division ohne Rest, und mod liefert diesen Rest.

a) Man berechne die Werte von f (n) für folgende Werte von n :

8, 11, 15, 16, 19, 83, 517

b) Man beweise: Für $0 \leq m, k \leq 7$ gilt

$$f(8m+k) = f(m+k) = \begin{cases} m+k & \text{für } m+k \leq 7 \\ m+k-7 & \text{sonst} \end{cases}$$

c) Man beweise (mit vollständiger Induktion): Für

$$n = z_k \cdot 8^k + z_{k-1} \cdot 8^{k-1} + \ldots\ldots + z_1 \cdot 8 + z_0 \quad \text{mit } z_i \in \{0, 1, \ldots, 7\} \text{ gilt}$$
$$f(n) = f(f(f(\ldots\ldots f(z_k + z_{k-1}) + z_{k-2}) + \ldots\ldots + z_1) + z_0)$$

d) Man gebe ein iteratives Programm (in einer abstrakten Programmnotation) an, das f (n) aus der Oktaldarstellung $z_k\ z_{k-1}\ z_{k-2} \ldots z_1\ z_0$ von n berechnet.

Man nehme an, die Ziffern der Oktaldarstellung liegen in einem Feld z[0], z[1], ... z[k-1], z[k] vor und k sei ebenfalls gegeben. Man verwende b) und c) und begründe damit auch die Korrektheit des Programms!

Thema Nr. 2

Teilaufgabe 1

Gegeben sei die Grammatik Γ mit der Menge $\{a, b\}$ von Terminalzeichen, der Variablenmenge $\{S, A, B, C\}$, dem Startsymbol S und den Produktionsregeln

$$S \rightarrow aA \mid bC \mid \varepsilon$$
$$A \rightarrow aA \mid bB \mid bC \mid \varepsilon$$
$$B \rightarrow bC \mid \varepsilon$$
$$C \rightarrow aA$$

(ε bezeichne das leere Wort.) $\mathcal{L}(\Gamma)$ sei die von Γ erzeugte Sprache.

a) Geben Sie einen nicht-deterministischen endlichen Automaten an, der genau die Elemente von $\mathcal{L}(T)$ akzeptiert.

b) Beweisen oder widerlegen Sie:

b1) $baabba \in \mathcal{L}(\Gamma)$

b2) Es gibt ein Wort von $\mathcal{L}(\Gamma)$, das mit bb beginnt.

b3) Kein Wort von $\mathcal{L}(\Gamma)$ enthält bbb als Teilwort.

Teilaufgabe 2

Sei $L = \{a^i b^j \mid i, j \in \mathbb{N}_0, i + 1 \leq j\}$ eine Sprache über dem Alphabet $\{a, b\}$.

a) Geben Sie eine kontextfreie Grammatik an, die L erzeugt!

b) Ist L regulär? Beweisen Sie Ihre Antwort!

Teilaufgabe 3

Sei $L = \{1^i 0 1^j 0 1^{i+j} \mid i, j \in \mathbb{N}_0\}$ eine Sprache über dem Alphabet $\{0, 1\}$.

a) Zeigen Sie, dass L entscheidbar ist. Geben Sie dazu eine deterministische Turingmaschine an, die für jedes $\omega \in \{0, 1\}^*$ entscheidet, ob $\omega \in L$ gilt oder nicht.

Erläutern Sie Ihre Lösung durch ausführliche Kommentare!

b) Erläutern Sie, was die Aussage

"L ist mit einer deterministischen Turingmaschine mit quadratischer Zeitkomplexität entscheidbar"

bedeutet, und begründen Sie, warum diese Aussage richtig ist!

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2003

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr**	
Kennwort: _____	**2003**	**46113**
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 3

Bitte wenden!

Thema Nr. 1

1. (Automatentheorie)
   a) Geben Sie einen deterministischen endlichen Automaten an, der genau diejenigen Wörter über $\{0,1,2\}$ akzeptiert, in denen weder 01 noch 10 vorkommt (d. h., die nicht die Form $u01v$ oder $u10v$ besitzen).
   b) Geben Sie einen regulären Ausdruck an, der die Menge dieser Wörter beschreibt!

2. (Formale Sprachen)
   a) Geben Sie eine kontextfreie Grammatik an, die die Sprache $\{a^n b^m c^{n+m}: n, m \geq 1\}$ erzeugt!
   b) Zeigen Sie mit Hilfe des Pumping-Lemmas, dass diese Sprache nicht regulär ist!

3. (Berechenbarkeit)
Es sei ϖ dasjenige Wort, das man aus $\omega \in \{0, 1\}^*$ durch das Vertauschen der Symbole 0 und 1 erhält, also z. B. $\overline{011011000100} = 100100111011$.
Man gebe eine 1-Band-Turingmaschine an, die die Funktion $f:\{0,1\}^* \to \{0,1\}^*$ mit

$$f(\omega) =_{def} \begin{cases} \overline{\omega} & \text{falls der rechteste Buchstabe von } \omega \text{ eine 1 ist} \\ \omega & \text{sonst} \end{cases}$$

berechnet. Die Arbeit der Turingmaschine beginnt und endet auf dem linkesten Buchstaben des Eingabe- bzw. Ergebniswortes. Kommentieren Sie die Befehle Ihrer Turingmaschine!

4. (Komplexität)
Geben Sie einen nichtdeterministischen Algorithmus an, der das *Rucksackproblem*

$$\left\{(a_1,\ldots,a_m,b_1\ldots,b_m,c,d):\ m,a_1,\ldots,a_m,b_1\ldots,b_m,c,d \in \mathbb{N} \wedge \exists I\left(I \subseteq \{1,\ldots,m\} \wedge \sum_{i\in I} a_i \leq c \wedge \sum_{i\in I} b_i \geq d\right)\right\}$$

in Polynomialzeit löst. Geben Sie eine Laufzeitabschätzung für Ihren Algorithmus an!

Alle Antworten sind zu begründen.

Thema Nr. 2

1. Durch folgende Grammatik (Großbuchstaben sind Variablen/Nichtterminalsymbole) wird eine „Programmiersprache" WHILE definiert:

   - $S \to W \mid \textit{while}\, B\, \textit{do}\, S\, \textit{end} \mid (S;S)$
   - $W \to V := V+1 \mid V := V-1$
   - $\mathrm{V} \to x\mathrm{K}$
   - $\mathrm{K} \to 0 \mid 1\mathrm{Z}$
   - $\mathrm{Z} \to 0\mathrm{Z} \mid 1\mathrm{Z} \mid \varepsilon$
   - $\mathrm{B} \to \mathrm{V} \neq 0$

a) Geben Sie einen regulären Ausdruck für die Sprache $\mathrm{LK} = \{\mathrm{w} \in \{0,1\}^* \mid \mathrm{w}$ ist aus K ableitbar$\}$ an!

b) Geben Sie einen endlichen erkennenden Automaten für $\mathrm{LK} = \{\mathrm{w} \in \{0,1\}^* \mid \mathrm{w}$ ist aus K ableitbar$\}$ an!

c) Ist die Sprache WHILE regulär, kontextfrei, kontextsensitiv? (Begründung!)

d) Geben Sie eine Beschreibung (Semantik) an, wie Programme aus WHILE auf einer geeigneten Maschine (beschreiben Sie auch diese!) ausgeführt werden!

e) Geben Sie ein Programm in WHILE an, das die Funktion $< x \to 2^x >$ berechnet!

2. Geben Sie eine formale Definition für eine zweidimensionale Turingmaschine, die anstelle eines (eindimensionalen) Turingbandes ein (zweidimensionales, wie auf kariertem Papier in Einheitsquadrate eingeteiltes) unendliches „Turing-Speicherblatt" besitzt!

3. Sei R ein regulärer Ausdruck und ε das leere Wort. Beweisen Sie:
   - Ist ε in der durch R beschriebenen regulären Sprache L(R), so gilt $\mathrm{R}^+ = \mathrm{R}^*$

   [Hinweis: $\mathrm{R}^+ = \mathrm{RR}^*$].

4. Gegeben sei die Sprache L= {Barbara, Bar, Barbar}.
   a) Geben Sie eine Grammatik G an, die L erzeugt!
   b) Geben Sie einen vollständigen erkennenden endlichen Automaten an, der L akzeptiert!

46113

Theoretische Informatik (nicht vertieft)

Herbst 2004

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst**	
Kennwort: _____	**2004**	**46113**
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 6

Bitte wenden!

Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1

Gegeben sei der nicht-deterministische endliche Automat M mit dem Eingabealphabet $\Sigma = \{a,b\}$, der Zustandsmenge $Q = \{q_0, q_1, q_2, q_3\}$ Anfangszustand q_0, Endzustand q_3 und der Übergangsfunktion δ mit:

$(q_0, a) \mapsto \{q_1\}$,
$(q_1, a) \mapsto \{q_1\}$,
$(q_1, b) \mapsto \{q_1, q_2\}$,
$(q_2, a) \mapsto \{q_3\}$,
$(q_2, b) \mapsto \{q_3\}$,
$(q, x) \mapsto \emptyset$ für alle übrigen $(q, x) \in Q \times \Sigma$.

L(M) sei die von *M* akzeptierte Sprache.

a) Beweisen oder widerlegen Sie:

a1) $aababb \in L(M)$.
a2) Jedes $w \in L(M)$ enthält das Teilwort *bb*.
a3) Das dritte Zeichen jedes $w \in L(M)$ ist ein *b*.
a4) Das vorletzte Zeichen jedes $w \in L(M)$ ist ein *b*.

b) Geben Sie eine reguläre (Typ-3-) Grammatik an, die *L(M)* erzeugt.

c) Konstruieren Sie aus *M* einen deterministischen endlichen Automaten, der *L(M)* akzeptiert.

d) Geben Sie einen regulären Ausdruck an, der *L(M)* beschreibt.

Fortsetzung nächste Seite!

Teilaufgabe 2

Die Sprache L über dem Alphabet $\{a,b,c\}$ sei gegeben durch $L = \{a^i b c^j \mid j > i \geq 0\}$.

a) Beweisen Sie: L ist nicht regulär.

b) Die Grammatik G mit den Terminalzeichen a, b, c, den Variablen S, A, C und dem Startzeichen S enthalte die Regeln

$$S \to bC \mid aACc$$
$$A \to \ldots$$
$$C \to \ldots$$

Vervollständigen Sie die Regeln für A und C so, dass G die Sprache L erzeugt!

c) Überführen Sie die Grammatik aus Teil b) in Chomsky-Normalform.

d) Ist die Sprache L mit einer deterministischen Turing-Maschine mit einer Zeitkomplexität $O(n^2)$ entscheidbar (n ist die Länge der jeweiligen Eingabe)? Begründen Sie Ihre Antwort!

Teilaufgabe 3

Sei Σ ein Alphabet.

a) Was bedeutet es, dass eine totale Funktion $f : \Sigma^* \to \Sigma^*$ mit einer Turing-Maschine berechenbar ist?

b) Für fest gewählte $u, v_1, v_2 \in \Sigma^*$ und eine totale Funktion $g : \Sigma^* \to \Sigma^*$ sei $h : \Sigma^* \to \Sigma^*$ definiert durch

$$h(w) = \begin{cases} v_1 & \text{falls } g(w) = u \\ v_2 & \text{sonst.} \end{cases}$$

Beweisen Sie: Ist g mit einer Turing-Maschine berechenbar, so gilt dies auch für h.

Thema Nr. 2

1. Automatentheorie

Es sei $\mathrm{bin}(n)$ die Binärdarstellung der natürlichen Zahl n (ohne führende Nullen).
Weiter sei $D =_{\mathrm{def}} \{\mathrm{bin}(n) : n \in \mathbb{N} \text{ und } n \equiv 5 \pmod 8\}$.

a) Geben Sie einen deterministischen endlichen Automaten an, der D akzeptiert!
b) Geben Sie einen regulären Ausdruck an, der D beschreibt!

2. Formale Sprachen

Es seien $M =_{\mathrm{def}} \{wuw^R v : w,u,v \in \{a,b\}^*\}$ und $N =_{\mathrm{def}} \{wcuw^R v : w,u,v \in \{a,b\}^*\}$.
Dabei sei das Zeichen $c \notin \{a,b\}$ und $(a_1 a_2 \ldots a_n)^R =_{\mathrm{def}} a_n \ldots a_2 a_1$ für alle $a_1, a_2, \ldots, a_n \in \{a,b\}$.

a) Geben Sie kontextfreie Grammatiken an, die M und N erzeugen.
b) Sind M und N regulär? (Begründung!)

Fortsetzung nächste Seite!

3. Endliche Automaten

Gegeben sei folgender nichtdeterministischer endlicher Automat A_a.

$$
\begin{aligned}
A_a &= \left(Q, \Sigma, \delta, q_0, \{q_3\}\right) \\
Q &= \{q_0, q_1, q_2, q_3\} \\
\Sigma &= \{a, b, c\} \\
\delta &= \{((q_0, a), q_2), ((q_0, a), q_1), \\
&\quad ((q_1, b), q_1), ((q_1, b), q_2), ((q_1, c), q_3), \\
&\quad ((q_2, a), q_2), ((q_2, a), q_1), ((q_2, c), q_3)\}
\end{aligned}
$$

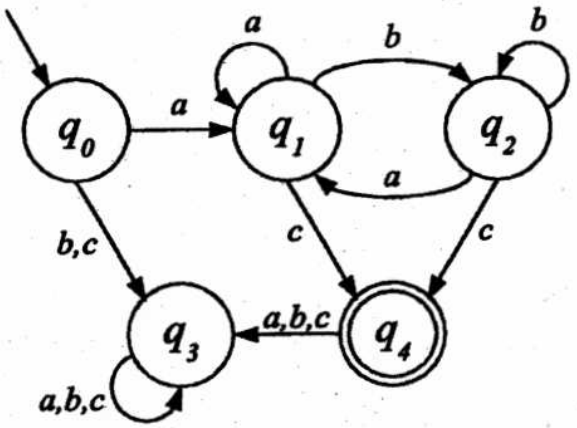
3.1 Stellen Sie den Automaten A_a graphisch dar!

3.2 Konstruieren Sie aus A_a einen per Konstruktion zu A_a äquivalenten deterministischen endlichen Automaten A_b! Stellen Sie den Automaten A_b graphisch dar!

3.3 Welche Sprache wird von A_b bzw. A_a akzeptiert (ohne Beweis)? Beschreiben Sie die Sprache durch einen regulären Ausdruck!

3.4 Geben Sie einen per Konstruktion minimalen und äquivalenten deterministischen endlichen Automaten $A_{c,\min}$ aus folgendem endlichen Automaten A_c! Stellen Sie den Automaten $A_{c,\min}$ graphisch dar!

$$
\begin{aligned}
A_c &= (Q, \Sigma, \delta, q_0, \{q_4\}) \\
Q &= \{q_0, q_1, q_2, q_3, q_4\} \\
\Sigma &= \{a, b, c\} \\
\delta &\subseteq (Q \times \Sigma) \times Q, \\
\delta &= \{((q_0, a), q_1), ((q_0, b), q_3), ((q_0, c), q_3), \\
&\quad ((q_1, a), q_1), ((q_1, b), q_2), ((q_1, c), q_4), \\
&\quad ((q_2, a), q_1), ((q_2, b), q_2), ((q_2, c), q_4), \\
&\quad ((q_3, a), q_3), ((q_3, b), q_3), ((q_3, c), q_3), \\
&\quad ((q_4, a), q_3), ((q_4, b), q_3), ((q_4, c), q_3)\}
\end{aligned}
$$



Fortsetzung nächste Seite!

4. Turing-Maschinen

Eine deterministische Turingmaschine ist ein Tupel $(Q, \Sigma, \Gamma, \delta, q_0, B, F)$. Dabei ist Q die endliche Zustandsmenge, Σ das endliche Eingabealphabet, Γ das endliche Bandalphabet, $\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$ die Übergangsfunktion, q_0 der Startzustand, B das Blanksymbol und F eine Menge von Endzuständen. Eine solche deterministische Turingmaschine liest also in jedem Schritt das aktuelle Zeichen unter dem Schreib-/Lesekopf, entscheidet abhängig vom aktuellen Zustand und dem gelesenen Zeichen, welches neue Zeichen geschrieben, in welchen Folgezustand übergegangen und ob dabei der Schreib-/Lesekopf nach rechts *(R)*, nach links *(L)* oder gar nicht *(N)* bewegt werden soll.

Sei nun $\Sigma = \{a, 0, 1\}$ und $\Gamma = \{a, 0, 1, B\}$. Konstruieren Sie eine Turingmaschine, welche eine Zahl in Binärdarstellung inkrementiert. Beachten Sie dabei die folgenden Vorgaben:

Auf dem Band steht als Eingabe ein a und rechts davon die gegebene Zahl in Binärdarstellung mit dem niederwertigsten Bit ganz rechts. Beispiel für die Dezimalzahl 13: $BBBa1101BBBBB$. Der Schreib-/Lesekopf steht zu Beginn der Verarbeitung auf dem Zeichen a und soll auch am Ende wieder auf diesem Zeichen a stehen. Das Zeichen a darf während einer ganzen Verarbeitung in keinem Fall überschrieben werden! Es darf auch kein weiteres Zeichen a geschrieben werden.

Schreiben Sie die Übergangsfunktion δ in Tabellenform nieder, pro Zustand eine Spalte, pro Zeile ein Bandsymbol und dann in jeder Zelle den Folgezustand, das zu schreibende Zeichen sowie die Kopfbewegung.

46113

Theoretische Informatik (nicht vertieft)

Herbst 2005

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	Herbst	46113
Kennwort: _____	2005	
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 4

Bitte wenden!

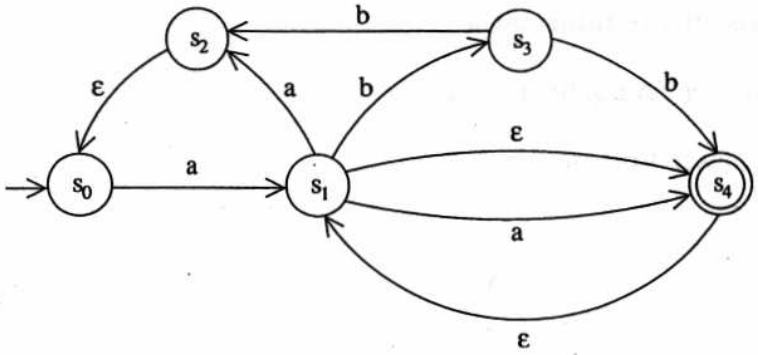
Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten!

Vorbemerkung: Bei der Lösung von Teilaufgaben können (wenn möglich) Lösungen der vorherigen Teilaufgaben vorausgesetzt werden, auch wenn diese nicht gelöst wurden.

Aufgabe 1

Gegeben ist der folgende nichtdeterministische endliche Automat A mit Leerübergängen. (Ein Leerübergang hat die Markierung "ε" und besagt, dass der Automat (spontan) einen Zustandswechsel durchführen kann ohne ein Eingabezeichen zu lesen.)



Es bezeichne $L(A)$ die von A akzeptierte Sprache.

a) Konstruieren Sie einen äquivalenten Automaten ohne Leerübergänge.

b) Konstruieren Sie einen äquivalenten, minimalen deterministischen Automaten.

c) Beweisen Sie unter Verwendung des in Teil b) konstruierten minimalen Automaten, dass für jedes $w \in L(A)$ gilt: Die Anzahl der in w vorkommenden Zeichen b ist gerade.

d) Geben Sie einen regulären Ausdruck an, der die Sprache $L(A)$ beschreibt.

e) Geben Sie eine rechtslineare Grammatik an, die die Sprache $L(A)$ erzeugt.

Aufgabe 2

Für eine primitiv rekursive Funktion $f : N \to N$ sei die (von f abhängige) Menge $M_f \subseteq N$ folgendermaßen definiert (wobei N die Menge der natürlichen Zahlen einschließlich der 0 bezeichnet):

$$M_f = \{x \in N \mid \exists z \leq x, f(z) = x\}$$

a) Geben Sie primitiv rekursive Funktionen $f1$ und $f2$ an mit $M_{f1} = N$ und $M_{f2} = \{0\}$.

b) Zeigen Sie, dass es keine primitiv rekursive Funktion f gibt mit $M_f = \emptyset$.

Es soll nun bewiesen werden, dass M_f primitiv rekursiv ist, d.h. dass die charakteristische Funktion $\chi_{M_f} : N \to N$ (mit $\chi_{M_f}(x) = 1$ falls $x \in M_f$, $\chi_{M_f}(x) = 0$ sonst) primitiv rekursiv ist. Zum Beweis sind die Teilaufgaben c) und d) zu bearbeiten.

c) Die (von f abhängige) Funktion $\Phi_f : N^2 \to N$ wird definiert durch

$$\Phi_f(x, y) = \begin{cases} 1 & \text{falls } \exists z \leq y \text{ mit } f(z) = x \\ 0 & \text{sonst} \end{cases}$$

Zeigen Sie, dass Φ_f primitiv rekursiv ist. Beim Beweis ist das Schema der primitiven Rekursion mit geeigneten primitiv rekursiven Funktionen $g : N \to N$ und $h : N^3 \to N$ zu verwenden. Dabei kann vorausgesetzt werden, dass die Funktionen $eq : N^2 \to N$ und $or : N^2 \to N$ primitiv rekursiv sind, wobei $eq(x, y) = 1$ falls $x = y$, $eq(x, y) = 0$ sonst und $or(x, y) = 1$ falls $x \neq 0$ oder $y \neq 0$, $or(x, y) = 0$ sonst.

d) Zeigen Sie unter Verwendung von Teil c), dass M_f primitiv rekursiv ist.

Thema Nr. 2

Aufgabe 1

Gegeben sind die drei folgenden Sprachen L_1, L_2 und L (jeweils über dem Alphabet $\{a,b,c\}$).

$$L_1 = \{a^i b^k c^k \mid i,k > 0\}$$
$$L_2 = \{a^i b^i c^k \mid i,k > 0\}$$
$$L = \{a^i b^j c^k \mid i,j,k > 0, i = j \text{ oder } j = k\}$$

a) Beweisen Sie, dass L_1 nicht regulär ist!

b) Geben Sie kontextfreie Grammatiken G_1 und G_2 an, die die Sprachen L_1 bzw. L_2 erzeugen!

c) Wie kann man aus den beiden Grammatiken für L_1 und L_2 eine Grammatik G konstruieren, die die Sprache L erzeugt? (Begründung ohne Beweis!)

d) Zeigen Sie, dass die in Teil c) konstruierte Grammatik G mehrdeutig ist!

e) Ist die Sprache $L_1 \cap L_2$ kontextfrei? (Begründung ohne Beweis!)

Aufgabe 2

Geben Sie einen vollständigen deterministischen endlichen Automaten (etwa in Form eines Zustandsdiagramms) an, der genau diejenigen Worte aus $\{a,b\}^*$ akzeptiert, die an drittletzter Position das Zeichen a stehen haben. Begründen (beweisen) Sie, dass der Automat alle diese und nur diese Worte akzeptiert!

Hinweis: Konstruieren Sie zuerst einen nichtdeterministischen endlichen Automaten, der die genannte Menge akzeptiert.

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2006

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr**	**46113**
Kennwort: _____	**2006**	
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 3

Thema Nr. 1

1. Betrachten Sie über dem Alphabet $\sum = \{a,b\}$ die Sprachen

$$L = \{w \in \sum^* ; w \text{ enthält das Teilwort } aa \text{ mindestens einmal } \}$$

$$M = \{w \in \sum^* ; w \text{ enthält eine gerade Anzahl von } a\text{'s } \}$$

a) Geben Sie für jede der Sprachen L und M einen vollständigen deterministischen endlichen Automaten an, der diese Sprache akzeptiert.

b) Geben Sie für die Sprachen L und M, sowie für deren Komplemente $\overline{L} = \sum^* \setminus L$ und $\overline{M} = \sum^* \setminus M$, jeweils einen regulären Ausdruck an, der diese Sprache beschreibt.

c) Konstruieren Sie einen vollständigen deterministischen endlichen Automaten, der die Sprache $L \cap M$ akzeptiert.
Hinweis: Verwenden Sie die Automaten aus Teilaufgabe a).

Fortsetzung nächste Seite!

2. Unter der (binären) Länge $l(n)$ einer positiven natürlichen Zahl versteht man die Anzahl der Ziffern in der Binärdarstellung von n, wobei führende Nullen weggelassen werden, also beispielsweise $l(23) = 5$, da 10111 die Binärdarstellung der Dezimalzahl 23 ist. Per Konvention setzt man $l(0) = 1$.

   a) Konstruieren Sie ein LOOP-Programm zur Berechnung der Längenfunktion $n \to l(n)$.

   b) Geben Sie eine primitiv-rekursive Definition der Längenfunktion $n \to l(n)$ an.

3. Ein Übungsleiter zur Algorithmik-Vorlesung möchte sich die Korrekturarbeiten zu den Programmieraufgaben erleichtern. Er gibt an, dass er ein Programm TESTER geschrieben habe, welches Folgendes leistet: Zu jeder Aufgabe vergleicht TESTER den Programmtext studentischer Lösungen mit einer Musterlösung des Übungsleiters und stellt fest, ob in beiden Fällen dieselbe Funktion berechnet wird.
   Kommentieren Sie diese Behauptung!

4. Betrachten Sie die beiden folgenden Entscheidungsprobleme:

   - TEILMENGENSUMME
     - Instanzen sind Paare $\langle (a_1,\ldots,a_n),k \rangle$ bestehend aus Vektoren positiver natürlicher Zahlen $(a_1,\ldots,a_n) \in \mathbb{N}_+^n$ und Zahlen $k \in \mathbb{N}$, wobei $n \geq 1$.
     - Gefragt ist, ob es eine Teilmenge $I \subseteq \{1,2,\ldots,n\}$ gibt mit $\sum_{i \in I} a_i = k$.

   - PARTITIONIERUNG
     - Instanzen sind Vektoren positiver natürlicher Zahlen $(a_1,\ldots,a_n) \in \mathbb{N}_+^n$ mit $n \geq 1$.
     - Gefragt ist, ob es eine Teilmenge $I \subseteq \{1,2,\ldots,n\}$ gibt mit $\sum_{i \in I} a_i = \sum_{j \notin I} a_j$.

   a) Zeigen Sie, dass die Abbildung

   $$\langle (a_1,\ldots,a_n),k \rangle \to \left(a_1,\ldots,a_n, 1-k+\sum_{i=1}^{n} a_i, k+1 \right)$$

   eine polynomielle Reduktion von TEILMENGENSUMME auf PARTITIONIERUNG ist.

   b) Was folgt daraus für die Zeitkomplexität von PARTITIONIERUNG?

Thema Nr. 2

1. Geben Sie einen vollständigen deterministischen endlichen Automaten an, der die Zeichenketten über $\{a,b,c\}$ mit einer geraden Anzahl von a's als Sprache hat.

2. Zeigen Sie, dass der Schnitt kontextfreier Sprachen nicht kontextfrei zu sein braucht.

3. Zeigen Sie, dass $f: N \times N \to N$: mit

$$f(x,y) = \begin{cases} x-y & \text{falls } x > y \\ o & \text{sonst} \end{cases}$$

primitiv rekursiv ist.

4. Leiten Sie die Zeitkomplexität von Mergesort her.
(Mergesort(Feld)

Ist |Feld| > l, so
{mergesort (Erste Hälfte des Feldes);
mergesort (Zweite Hälfte des Feldes);
verschmelze die beiden Hälften
}
gebe das Feld zurück)

46113

Theoretische Informatik (nicht vertieft)

Herbst 2006

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst**	**46113**
Kennwort: _____	**2006**	
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

- Prüfungsaufgaben -

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

Bitte wenden!

Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten!

Teilaufgabe 1

Gegeben sei der nicht-deterministische endliche Automat M mit dem Eingabealphabet $\Sigma = \{a,b\}$, der Zustandsmenge $Q = \{q_0, q_1, q_2, q_3\}$, Anfangszustand q_0, Endzustand q_3 und der Übergangsfunktion δ mit:

$$(q_0, a) \mapsto \{q_1\},$$
$$(q_0, b) \mapsto \{q_2\},$$
$$(q_1, b) \mapsto \{q_1, q_2, q_3\},$$
$$(q_2, a) \mapsto \{q_3\},$$
$$(q_3, a) \mapsto \{q_3\},$$
$$(q, x) \mapsto \emptyset \quad \text{für alle übrigen } (q,x) \in Q \times \Sigma$$

$L(M)$ sei die von M akzeptierte Sprache.

a) Beweisen Sie:
   a1) $abbba \in L(M)$
   a2) In jedem $w \in L(M)$ kommt a mindestens einmal vor.
   a3) Zu jedem $n \in \mathbb{N}$ gibt es ein $w \in L(M)$, in dem a mehr als n-mal vorkommt.

b) Geben Sie eine reguläre (Typ-3-) Grammatik an, die $L(M)$ erzeugt.

c) Konstruieren Sie aus M einen deterministischen endlichen Automaten, der $L(M)$ akzeptiert.

d) Geben Sie einen regulären Ausdruck an, der $L(M)$ beschreibt.

Fortsetzung nächste Seite!

Teilaufgabe 2

Gegeben seien das Alphabet $\Sigma = \{a,b\}$, die Grammatik $G = (\{S,A\},\Sigma,S,P)$ mit der aus den Produktionsregeln

$$S \to Ab$$

$$A \to b \,\Big|\, aAa$$

bestehenden Menge P sowie die Sprache $L = \left\{a^n b a^n b \,|n \geq 0\right\}$ über Σ.

a) Beweisen Sie: L ist die von G erzeugte Sprache.

b) Beweisen Sie: L ist nicht regulär.

c) Überführen Sie G in Chomsky-Normalform.

Teilaufgabe 3

Es seien Σ ein Alphabet, a ein Zeichen von Σ und L_1 und L_2 zwei Sprachen über Σ.
Gelten folgende Aussagen? Begründen Sie Ihre Antworten.

a) Sind L_1 und L_2 semi-entscheidbar, so ist $L_1 \setminus L_2$ semi-entscheidbar.

b) Sind L_1 und L_2 entscheidbar, so ist die Funktion $f : \Sigma^* \to \Sigma^*$ mit

$$f(w) = \begin{cases} a & \text{falls } w \in L_1 \cap L_2 \\ aa & \text{sonst} \end{cases}$$

berechenbar.

c) Ist L_1 mit einer deterministischen Turing-Maschine mit einer Zeitkomplexität $O(n)$ entscheidbar, so gilt dies auch für $\Sigma^* \setminus L_1$.
(n ist die Länge der jeweiligen Eingabe.)

d) Sind sowohl L_1 als auch L_2 mit einer deterministischen Turing-Maschine mit einer Zeitkomplexität $O(n^2)$ entscheidbar, so gilt dies auch für $L_1 \cup L_2$.
(n ist die Länge der jeweiligen Eingabe.)

Thema Nr. 2

Aufgabe 1 (reguläre Sprachen und endliche Automaten)

Die Elemente einer regulären Sprache können durch deterministische oder nicht-deterministische endliche Automaten erkannt werden.
Betrachten Sie folgenden nicht-deterministischen endlichen Automaten
$A_1 = \left(\{q_1, q_2, q_3, q_4, q_5\}, \{0,1\}, \delta, q_1, \{q_4\}\right)$ mit Zustandsmenge $\{q_1, q_2, q_3, q_4, q_5\}$ Eingabealphabet $\{0,1\}$, Anfangszustand q_1 und Endzustandsmenge $\{q_4\}$. Die Übergangsfunktion δ sei durch folgende Tabelle definiert:

δ	q_1	q_2	q_3	q_4	q_5
0	$\{q_1, q_2\}$	$\{q_3, q_4\}$	$\{q_1, q_3\}$	$\{q_5\}$	$\{q_5\}$
1	$\{q_1\}$	$\{q_4\}$	$\{q_4\}$	$\{q_5\}$	$\{q_5\}$

a) Zeichnen Sie das Übergangsdiagramm des Automaten mit Zuständen und Übergangskanten.

b) Beschreiben Sie die von A_1 erkannte reguläre Sprache L_1, indem Sie eine mathematisch exakte Definition der Menge der erkannten Worte über $\{0,1\}$ angeben. Begründen Sie Ihre Antwort.

c) Geben Sie einen möglichst kurzen regulären Ausdruck an, der die Sprache L_1 beschreibt.

d) Wandeln Sie den nicht-deterministischen endlichen Automaten A_1 in einen deterministischen endlichen Automaten A_2 um, indem Sie die Teilmengenkonstruktion anwenden. Konstruieren Sie dazu ausgehend vom Anfangszustand von A_1 die ε-Folgezustände der jeweils entstehenden Zustände. Geben Sie für A_2 sowohl ein Übergangsdiagramm als auch eine tabellenförmige Darstellung der Übergangsfunktion an.

e) Definieren Sie die Äquivalenz von Zuständen in endlichen Automaten.

f) Bestimmen Sie mit Hilfe des Table-Filling-Verfahrens alle äquivalenten Zustände von A_2. Bauen Sie dazu die vollständige Tabelle mit Zustandspaaren schrittweise auf und markieren Sie, ob die jeweiligen Zustände unterscheidbar sind. Erläutern Sie jeden durchgeführten Schritt. Fassen Sie anschließend die äquivalenten Zustände zusammen und konstruieren Sie den resultierenden deterministischen endlichen Automaten A_3, indem Sie für A_3 ein Übergangsdiagramm und eine tabellenförmige Darstellung der Übergangsfunktion angeben.

g) Gibt es einen deterministischen endlichen Automaten mit weniger Zuständen als A_3, der die reguläre Sprache L_1 erkennt? Begründen Sie Ihre Antwort kurz. Geben Sie gegebenenfalls einen deterministischen endlichen Automaten mit weniger Zuständen als A_3 an.

Fortsetzung nächste Seite!

Aufgabe 2 (kontextfreie Sprachen und Kellerautomaten)

a) Betrachten Sie die kontextfreie Sprache $L = \left\{a^n b^n ; n \geq 1\right\}$ über dem Alphabet $\{a,b\}$.
Geben Sie eine kontextfreie Grammatik G mit Terminalsymbolen, Nichtterminalsymbolen und Produktionen an, die L erzeugt.

b) Konstruieren Sie einen nicht-deterministischen Kellerautomaten $K = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, der L erkennt. Geben Sie eine genaue Definition aller Elemente des Kellerautomaten mit einer mathematisch exakten Definition der Übergangsrelation δ an. Erläutern Sie die Arbeitsweise des Kellerautomaten und begründen Sie, warum K alle Worte aus L erkennt.

c) Erläutern Sie den Unterschied zwischen nicht-deterministischen und deterministischen Kellerautomaten durch Angabe der exakten Definitionen. Welche Unterschiede in den Verarbeitungsschritten gibt es?

d) Kann die Sprache $L = \left\{a^n b^n ; n \geq 1\right\}$ durch einen deterministischen Kellerautomaten erkannt werden? Begründen Sie Ihre Antwort.

e) Betrachten Sie die folgende kontextfreie Grammatik $G = (\{S\}, \{0,1,+,*\}, P, S)$ mit den Produktionen $P = \left\{S \to S + S, S \to S * S, S \to 0, S \to 1\right\}$. Beweisen Sie, dass diese Grammatik mehrdeutig ist.

Aufgabe 3 (Berechenbarkeit und Turingmaschinen)

a) Konstruieren Sie eine deterministische Turingmaschine zum Erkennen der Sprache $L = \left\{0^n 1^n ; n \geq 1\right\}$ über dem Alphabet $\{0,1\}$. Beim Start der Turingmaschine stehe das Eingabewort $w \in \{0,1\}^*$ auf dem Band. Erläutern Sie die Rolle der Zustände der von Ihnen konstruierten Turingmaschine und geben Sie die Übergangsfunktion in Tabellenform an.

b) Illustrieren Sie die Arbeitsweise der von Ihnen konstruierten Turingmaschine, in dem Sie die Berechnungsschritte für die Eingabe 0011 als Konfigurationsübergänge angeben.

c) Erläutern Sie die Funktionsweise nichtdeterministischer Turingmaschinen. Erkennen nicht-deterministische Turingmaschinen dieselbe Sprachklasse wie deterministische Turingmaschinen? Geben Sie eine ausführliche Begründung für Ihre Antwort.

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2007

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr 2007**	**46113**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 4

Bitte wenden!

Thema Nr. 1

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1:
(Automatentheorie)

Gegeben seien die Sprachen

$$\begin{aligned} L_1 &= \{(ab)^i c\, d^j : i,j \in \mathbb{N}\}, \\ L_2 &= \{a^i\, b^j\, c^i\, d^j : i,j \in \mathbb{N}\}, \\ L_3 &= \{a^i\, b^j (cd)^j : i,j \in \mathbb{N}\}. \end{aligned}$$

Für welche Sprache existiert ein endlicher Automat, der diese erkennt? Falls es einen solchen Automaten gibt, geben Sie ihn an! Andernfalls begründen Sie die Nichtexistenz!

Aufgabe 2:
(Formale Sprachen)

Zeigen Sie, dass kontextfreie Sprachen nicht abgeschlossen bzgl. der Durchschnittsbildung sind! Betrachten Sie hierzu

$$\begin{aligned} L_1 &= \{a^n\, b^n\, c^m : n,m \in \mathbb{N}\} \\ L_2 &= \{a^m\, b^n\, c^n : n,m \in \mathbb{N}\} \end{aligned}$$

Zeigen Sie, dass L_1 und L_2 kontextfrei sind, nicht aber $L_1 \cap L_2$ (Pumping Lemma).

Aufgabe 3:
(Berechenbarkeit und Komplexität)

Gegeben sei die Sprache $L = \{a^n\, b\, c^n : n \in \mathbb{N}\}$.

a) Geben Sie ein Turing-Programm an, das für jedes $w \in \{a,b,c\}^*$ entscheidet, ob $w \in L$ gilt oder nicht. Beschreiben Sie jeden Schritt im Detail!

b) Welche Komplexität besitzt der Algorithmus?

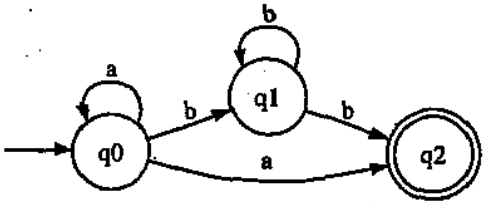
Thema Nr. 2

Sämtliche Teilaufgaben sind zu bearbeiten!

Aufgabe 1:

(Automatentheorie)

Gegeben sei der folgende nichtdeterministische Automat N.



a) Geben Sie einen regulären Ausdruck für die von N erkannte Sprache an!

b) Wandeln Sie N in einen deterministischen endlichen Automaten D um! Gehen Sie dabei systematisch vor und beschreiben Sie jeden der Schritte!

Aufgabe 2:

(Formale Sprache)

Hinweis:
Eine Chomsky-Normalform ist eine kontextfreie Grammatik G mit $\epsilon \notin L(G)$, welche nur Regeln der Form

$$A \rightarrow BC$$
$$bzw.\ A \rightarrow a$$

besitzt.
(A, B, C Variablen; a Terminsymbol)

a) Beweisen Sie: Wenn eine Grammatik G in Chomsky-Normalform ist, dann wird ein Wort w der Sprache $L(G)$ in genau $2|w| - 1$ Ableitungsschritten erzeugt.

b) Gegeben sei die folgende Grammatik für aussagenlogische Konjunktionen, die nur die Aussagenvariablen „A" und „B" enthalten.

$$Conj \rightarrow Lit \mid Lit\ '\wedge'\ Conj$$
$$Lit \rightarrow Term \mid '\neg'\ Term$$
$$Term \rightarrow '\ A'\ \mid 'B'$$

Beispiele, die mit dieser Grammatik generiert werden können, sind:

$$\neg A$$
$$B$$
$$A \wedge \neg B$$
$$A \wedge B \wedge A \wedge B$$

Wandeln Sie diese Grammatik in Chomsky-Normalform um!

Fortsetzung nächste Seite!

Aufgabe 3:
(Berechenbarkeit)

a) Definieren Sie den Begriff „rekursiv aufzählbar" für Sprache $A \subseteq \sum^*$!

b) Zeigen Sie: Sind die Sprachen A und B rekursiv aufzählbar, so auch $A \cup B$ und $A \times B$.

Aufgabe 4:
(Komplexität)

Der folgende Algorithmus (in Java) sortiert eine Reihung von ganzen Zahlen.

```
public static void selectionSort(int data[]) {
    for (int i = 0; i < (data.length-1); i++) {
        // Finde das Minimum
        int minIndex = i;
        for (int j = i + 1; j < data.length; j++) {
            if (data[j] < data[minIndex]) {
                minIndex = j;
            }
        }
        // Vertausche aktuelles Datum und Minimum
        int tmp = data[i];
        data[i] = data[minIndex];
        data[minIndex] = tmp;
    }
}
```

Welche Zeitkomplexität hat `selectionSort`? Berechnen Sie genau die Anzahl der benötigten Vergleiche und folgern Sie daraus die Komplexität in O-Notation!

46113

Theoretische Informatik (nicht vertieft)

Herbst 2007

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____		
Kennwort: _____	Herbst 2007	46113
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

Bitte wenden!

Thema Nr. 1

Es sind alle Teilaufgaben zu bearbeiten!

Aufgabe 1 (endliche Automaten)
Gegeben sei ein nichtdeterministischer endlicher Automat A_1 mit Zustandsmenge $Q = \{A, B, C, D, E\}$, Eingabealphabet $\sum = \{0, 1\}$, Startzustand A und Endzustandsmenge $F = \{E\}$.
Die Übergangsfunktion $\delta : Q \times \sum \cup \{\epsilon\} \rightarrow P(Q)$ sei durch folgende Tabelle gegeben:

δ	A	B	C	D	E
0	$\{C\}$	$\emptyset$	$\{A\}$	$\{E\}$	$\{D\}$
1	$\{C\}$	$\{B\}$	$\{D\}$	$\{A\}$	$\{E\}$
ϵ	$\{B\}$	$\emptyset$	$\emptyset$	$\{B\}$	$\emptyset$

Dabei bezeichne $P(Q)$ die Potenzmenge von Q.

a) Zeichnen Sie den Automaten in Form eines Übergangsdiagramms, das für jeden Zustand einen Knoten und Kanten zwischen den Knoten gemäß der Übergangsfunktion enthält. Markieren Sie den Endzustand gesondert.

b) Konstruieren Sie mit Hilfe der Teilmengenkonstruktion einen zu A_1 äquivalenten deterministischen endlichen Automaten A_2. Erzeugen Sie dazu vom Startzustand von A_1 ausgehend schrittweise die Zustände des deterministischen endlichen Automaten A_2 als Teilmengen der Zustände von A_1. Geben Sie die jeweils erzeugten Teilmengen an. Zeichnen Sie den so erhaltenen Automaten A_2 als Übergangsdiagramm.

c) Identifizieren Sie äquivalente Zustände von A_2, indem Sie den Table-Filling-Algorithmus anwenden. Bauen Sie dazu schrittweise eine Tabelle auf, in der für die möglichen Zustandspaare vermerkt ist, ob sie unterscheidbar sind. Begründen Sie die Schritte des Algorithmus.

d) Konstruieren Sie einen zu A_2 äquivalenten deterministischen endlichen Automaten, der eine minimale Anzahl von Zuständen enthält. Fassen Sie dazu alle äquivalenten Zustandspaare von A_2 zu jeweils einem Zustand zusammen. Zeichnen Sie den resultierenden Automaten als Übergangsdiagramm.

Fortsetzung nächste Seite!

Aufgabe 2:

(Wortproblem für kontextfreie Grammatiken)

Gegeben sei folgende kontextfreie Grammatik in Chomsky-Normalform:

$$G = (\{A, B, C, S\}, \{a, b\}, P, S)$$

mit Variablenmenge $V = \{A, B, C, S\}$, Alphabet $\sum = \{a, b\}$ und Startsymbol S. Die Produktionen P sind gegeben durch:

$$\begin{aligned} P = \{ \; & S \rightarrow AA \mid BC \\ & A \rightarrow BA \mid a \\ & B \rightarrow CB \mid b \\ & C \rightarrow AB \mid a \; \} \end{aligned}$$

a) Beweisen Sie, dass diese Grammatik mehrdeutig ist, d. h. dass es für mindestens ein Wort der von G erzeugten Sprache mindestens zwei unterschiedliche Ableitungsbäume gibt.

b) Entscheiden Sie durch Anwendung des CYK-Algorithmus, ob das Wort $w = baab$ von der Grammatik G erzeugt wird. Konstruieren Sie dazu schrittweise eine Tabelle, deren Einträge X_{ij} alle Variablen der Grammatik enthält, die das Teilwort $a_i \ldots a_j$ des gegebenen Wortes $w = a_1 \ldots a_n$ erzeugen können. Geben Sie die zur Ermittlung jedes Eintrages erforderlichen Berechnungen an und begründen Sie die Berechnungen.

Aufgabe 3:

(Berechenbarkeit und Turingmaschinen)

a) Konstruieren Sie eine determinstische Turingmaschine, die eine Eingabe $x \in \{0, 1\}^*$ als Binärzahl interpretiert und zu x die Zahl 1 addiert. Beim Start der Turingmaschine stehe das Eingabewort x auf dem Eingabeband. Beschreiben Sie das Vorgehen und die Arbeitsweise der von Ihnen entworfenen Turingmaschine. Erläutern Sie die Rolle der Zustände der von Ihnen entworfenen Turingmaschine und geben Sie die Übergangsfunktion in Tabellenform an. Zeichnen Sie die Turingmaschine als Übergangsdiagramm.

b) Illustrieren Sie die Arbeitsweise der von Ihnen konstruierten Turingmaschine, indem Sie die Berechnungsschritte der Turingmaschine für die Eingabe 101 angeben. Verwenden Sie dazu die Beschreibung durch Konfigurationsübergänge.

c) Turingmaschinen können dadurch erweitert werden, dass das Eingabeband in mehrere Spuren unterteilt wird. Geben Sie an, ob durch diese Erweiterung die Möglichkeit des Beschreibungsmechanismus erweitert wird, d. h. beantworten Sie die Frage, ob durch die Erweiterung Sprachen erkannt werden können, die ohne die Erweiterung nicht erkannt werden können. Begründen Sie Ihre Antwort entweder dadurch, dass Sie eine solche Sprache angeben oder einen Mechanismus skizzieren, der die Simulation von Turingmaschinen mit mehreren Spuren auf eine Turingmaschine mit einer Spur durchführt.

Thema Nr. 2

Aufgabe 1:
(formale Sprache)

Sei L die Menge aller Worte über dem Alphabet $\{a, b\}$,

deren zweites Zeichen (von links) ein a,
deren drittletztes Zeichen ein b ist und
bei denen nie zwei a´s aufeinander folgen.

Ist L regulär oder kontext-frei und nicht regulär oder nicht kontextfrei?
Begründen Sie Ihre Antwort durch die Angabe einer passenden Grammatik oder eines Automaten.

Aufgabe 2:
(Abschlusseigenschaften)

Zeigen Sie:

i) Die regulären Sprachen sind abgeschlossen unter Durchschnitt.

ii) Die kontextfreien Sprachen sind nicht abgeschlossen unter Durchschnitt.

Begründen Sie Ihre Antwort bei Teilaufgabe i durch die Konstruktion eines entsprechenden Automaten bzw. einer Grammatik ohne den Beweis der Korrektheit ihrer Konstruktion.

Aufgabe 3:
(kontextfreie Sprachen)

Definition:

Eine kontextfreie Grammatik $G = (N, T, P, S)$, die nur Worte der Länge ≥ 3 erzeugt, ist in 3-Normalform, wenn die Produktionen die Form

$$A \to BCB \quad \text{oder} \quad A \to a \quad \text{haben}$$

mit nicht-terminalen A, B, C, D und terminalen Zeichen a.
(Die 2-Normalform ist die Chomsky Normalform.)

Zeigen Sie:

Zu jeder kontextfreien Sprache L, die nur Worte x der Länge ≥ 3 enthält, gibt es eine kontextfreie Grammatik G in 3-Normalform, die L erzeugt.
Geben Sie die Konstruktion an. Der Äquivalenz- oder Korrektheitsbeweis ist nicht verlangt. Hinweis: Sie dürfen die Chomsky Normalform verwenden.

Fortsetzung nächste Seite!

Aufgabe 4 (NP-Vollständigkeit)

Das k-Färbbarkeitsproblem auf Graphen ist das Problem, ob man die Knoten eines Graphen G so mit höchstens k Farben färben kann, dass benachbarte Knoten verschiedene Farben haben.

Warum ist das k-Färbbarkeitsproblem in NP?
(Es ist sogar bekannt, dass das k-Färbbarkeitsproblem NP-vollständig ist.)

Aufgabe 5 (Berechenbarkeit)

Was halten Sie von der These:
„Alles was exakt beschrieben werden kann, kann man auch berechnen.“

Geben Sie Argumente für oder gegen diese These an.

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2008

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____		
Kennwort: _____	Frühjahr 2008	46113
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik, Algorithmen**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

Bitte wenden!

Thema Nr. 1

Aufgabe 1:

reguläre Mengen

Sei L die Menge aller Worte über dem Alphabet $\{a, b, c\}$, deren drittes Zeichen (von links) ein a und deren vorletztes Zeichen ein b ist.

Beschreiben Sie L

a) durch einen regulären Ausdruck,

b) durch einen deterministischen endlichen Automaten A.

Aufgabe 2:

Klassifikation von Sprachen

Was ist kleinste Klasse der Chomsky-Hierarchie, in der die Sprache

$$L = \{a^n b^n c^n \mid 0 \leq n \leq 100\} \text{ liegt.}$$

Begründen/beweisen Sie Ihre Antwort!

Sie dürfen hierzu bekannte Eigenschaften von Klassen der Chomsky-Hierarchie verwenden, ohne diese selbst zu beweisen, z. B. Pumping Lemmata für reguläre bzw. kontextfreie Sprachen oder entsprechende Automaten oder Maschinen bzw. Grammatiken. Bei der Angabe von Automaten oder Maschinen müssen Sie „high-level" beschreiben, wie der Automat/die Maschine arbeiten soll.

Aufgabe 3:

Determinismus und Nicht-Determinismus

Gegeben sind

a) endliche Automaten,

b) Kellerautomaten und

c) Turingmaschinen.

Für welche dieser Maschinen gilt, dass die deterministischen und die nicht-deterministischen Maschinen gleichmächtig sind bzw. dass dies nicht gilt?

Begründen Sie Ihre Antwort für (beliebige) zwei der drei Fälle a) - c)! Im Fall der Gleichmächtigkeit müssen Sie zur Begründung angeben, wie eine nicht-deterministische Maschine durch eine äquivalente deterministische simuliert wird. Der Beweis der Äquivalenz ist nicht gefordert. Andernfalls ist der Nachweis anhand passender und bekannter Gegenbeispiele zu führen.

Fortsetzung nächste Seite!

Aufgabe 4:

Konstruktion von Turingmaschinen

a) Konstruieren Sie eine (k-Band-)Turingmaschine M für die Sprache

$$COPY = \{wcw | w \in \{a, b\}^*\}.$$

b) Welche Zeit-Komplexität hat Ihre Turingmaschine M?

c) Ordnen Sie $COPY$ in die Chomsky-Hierarchie ein.

Aufgabe 5:

kontextfreie Sprachen

Zeigen Sie durch Angabe einer kontextfreien Grammatik G, dass die Sprache L der Palindrome über dem Alphabet $\{a, b, c\}$

$$L = \{w \in \{a, b, c\}^* \mid w = w^{rev}\}$$

kontextfrei ist, wobei w^{rev} die Spiegelung von w ist (w wird rückwärts gelesen).

Aufgabe 6:

kontextfreie Sprachen

Zeigen Sie, dass die Sprache

$$L = \{a^n b^m | n \geq 0, m \geq 0, n = m \text{ oder } n = 2m\}$$

kontextfrei ist.

Es reicht die Angabe einer geeigneten Grammatik oder eines geeigneten Automaten. Was „geeignet“ ist, muss gesagt werden.

Thema Nr. 2

Aufgabe 1:

Primitiv rekursive Funktionen

Geben Sie für die folgenden Funktionen primitiv rekursive Terme an. In jeder Teilaufgabe können Sie die Funktionen der vorangegangenen Teilaufgaben als primitiv rekursiv voraussetzen.

a) Die Bedingung $if : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit

$$if(b, x, y) = \begin{cases} x \text{ falls } b > 0 \\ y \text{ sonst} \end{cases}$$

b) Die Vorgängerfunktion $pred : \mathbb{N} \to \mathbb{N}$ mit

$$pred(x) = \begin{cases} x - 1 \text{ falls } x > 0 \\ 0 \text{ sonst} \end{cases}$$

c) Die absolute Differenz $diff : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit

$$diff(x, y) = \begin{cases} x - y \text{ falls } x \geq y \\ y - x \text{ sonst} \end{cases}$$

d) Das Gleichheitsprädikat $eq : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ mit

$$eq(x, y) = \begin{cases} 1 \text{ falls } x = y \\ 0 \text{ sonst} \end{cases}$$

Aufgabe 2:

Typ-1-Sprachen

Gegeben sei die Sprache $L = \{a^k b^{2k} c^k \mid k \in \mathbb{N}\}$.

a) Zeigen Sie, indem Sie eine geeignete Grammatik angeben: L ist kontextsensitiv.

b) Verwenden Sie das Pumping-Lemma für kontextfreie Sprachen (uvwxy-Theorem), um zu zeigen: L ist *nicht* kontextfrei.

Fortsetzung nächste Seite!

Aufgabe 3:

Cocke-Younger-Kasami-Algorithmus

Es sei $G = (\{A, B, C, D, E, F, G, H, I\}, \{d, q, m, p, b\}, P, A)$ die kontextfreie Grammatik mit dem Startzustand A und den folgenden Regeln P in Chomsky-Normalform.

$$
\begin{aligned}
A &\rightarrow FB|FC \\
B &\rightarrow AG \\
C &\rightarrow DG \\
D &\rightarrow HE|HI \\
E &\rightarrow DI \\
F &\rightarrow d \\
G &\rightarrow b \\
H &\rightarrow q \\
I &\rightarrow p
\end{aligned}
$$

a) Zeigen Sie mit Hilfe des Algorithmus von Cocke, Younger und Kasami (CYK), dass $ddqpbb \in L(G)$.

b) Zeigen Sie mit Hilfe CYK-Algorithmus, dass $ddqqppb \notin L(G)$.

Aufgabe 4:

Kellerautomaten

Es sei $L = \{a^{2i}bc^i \mid i \in \mathbb{N}\}$

a) Geben Sie eine kontextfreie Grammatik G an, so dass $L(G) = L$.

b) Geben Sie einen nicht-deterministischen Kellerautomaten $M = (Q, \sum, \Gamma, \delta, q_0, \#)$ an, der die Sprache L akzeptiert, und erläutern Sie seine Funktionsweise.

46113

Theoretische Informatik (nicht vertieft)

Herbst 2008

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2008**	**46113**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **4**

Thema Nr. 1

Aufgabe 1

Ein Bauer will mit einem Wolf, einer Ziege und einem Kohlkopf einen Fluss überqueren. Das Boot ist klein, so dass er auf einer Überfahrt jeweils nur eines der Tiere oder den Kohlkopf mitnehmen kann. Ohne die Aufsicht des Bauern verschlänge der Wolf die Ziege und die Ziege fräße den Kohlkopf.

Modellieren Sie das Problem durch einen endlichen Automaten, der nur die Lösungen akzeptiert. Die Zustände beschreiben die Situationen auf beiden Seiten des Flusses, z. B. die Anfangssituation „{ BZWK}~ {}, wobei B den Bauern, W den Wolf, Z die Ziege, K den Kohlkopf und ~ den Fluss bezeichne. Die Eingaben stellen die Überfahrten dar; b, bw, bz, bk bezeichnen die Überfahrten des Bauern allein, mit dem Wolf, mit der Ziege, dem Kohlkopf.

Fortsetzung nächste Seite!

Aufgabe 2

Gegeben ist die Grammatik G mit der Variablen S, dem Startzeichen S, den Terminalzeichen a, b und c sowie Regeln

$$S \rightarrow aS|Sa|bSb|c$$

a) Geben Sie die Sprache L(G) in Mengenschreibweise an.
b) Bringen Sie G in Chomsky-Normalform.
c) Überprüfen Sie mit Hilfe des CYK-Algorithmus, ob das Wort abacab in der Sprache L(G) liegt. Geben Sie die Tabelle an.

Aufgabe 3

a) Definieren Sie das Halteproblem für Turing-Maschinen als Sprache H über dem Alphabet $\{0,1,\#\}$.

b) Gegeben sei das folgende Problem E:

> Entscheiden Sie, ob die deterministische Turing-Maschine codiert durch n auf genau den Eingaben 10 und 01 hält.

Zeigen Sie, dass E nicht entscheidbar ist. Benutzen Sie, dass H aus a) nicht entscheidbar ist.

c) Angenommen, es wurde gezeigt, dass P = NP ist. Zeigen Sie, dass dann *jede* Sprache $L \in P$ sogar NP-vollständig ist.

Aufgabe 4

Sei $\Sigma = \{0,1\}$, und sei $w \in \Sigma^*$. $\#_0(w)$ gibt an, wie oft die 0 in w vorkommt. Z. B. ist $\#_0(0110100) = 4$. Analog zählt $\#_1(w)$, die Anzahl der Ziffern 1 in w.

a) Sei

$$L_1 = \{w \mid w \in \{0,1\}^*, \#_0(w) = \#_1(w)\}.$$

Beispiele: $010110 \in L_1$, $1001 \in L_1$.
Zeigen Sie, dass L_1 kontextfrei ist, indem Sie die Arbeitsweise eines deterministischen Kellerautomaten M genau beschreiben, der L_1 akzeptiert. (Zu „genau" gehört auch die Begründung dafür, warum M Wörter, die nicht in L_1 sind, verwirft.)

b) Formulieren Sie das Pumping-Lemma für reguläre Sprachen:

„Sei L eine reguläre Sprache über dem Alphabet Σ. Dann gibt ..."

c) Zeigen Sie mittels Pumping-Lemma für reguläre Sprachen, dass die Sprache L_1 aus a) nicht regulär ist.

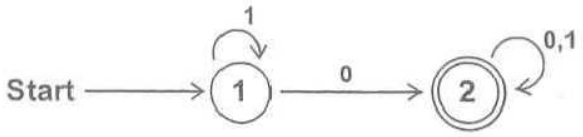
Thema Nr. 2

Aufgabe 1 (reguläre Ausdrücke und endliche Automaten)

Die Elemente einer regulären Sprache können durch endliche Automaten oder reguläre Ausdrücke beschrieben werden.

(a) Leiten Sie einen regulären Ausdruck her, der alle Worte über $\sum = \{0, 1\}$ beschreibt, die abwechselnd 0 oder 1 enthalten. Die Sprache sei L.
Beispiel: $0101 \in L, 1010 \in L, 0110 \notin L$
Berücksichtigen Sie alle Möglichkeiten, die Worte aus L zu bilden. Begründen Sie, warum der von Ihnen hergeleitete reguläre Ausdruck die Sprache L beschreibt.

(b) Betrachten Sie folgenden deterministischen endlichen Automaten



Welche reguläre Sprache wird von dem Automaten erkannt? Leiten Sie einen regulären Ausdruck ab, der die gleiche Sprache wie der angegebene Automat beschreibt. Bestimmen Sie dazu die regulären Ausdrücke $R_{ij}^{(k)}$, wobei $R_{ij}^{(k)}$ die Menge aller Worte w beschreibt, die der Beschriftung der Kanten auf einem Pfad zwischen Zustand q_i und q_j des Automaten entsprechen, wobei für keinen der Zwischenknoten q_l auf dem Pfad $l > k$ gelten darf. Vereinfachen Sie in den Zwischenschritten die erhaltenen regulären Ausdrücke so weit wie möglich. Welcher der Ausdrücke $R_{ij}^{(k)}$ beschreibt das Verhalten des gesamten Automaten? Begründen Sie, warum der von Ihnen hergeleitete reguläre Ausdruck die gleiche Sprache wie der angegebene Automat beschreibt.

(c) Geben Sie die Aussage des Pumping-Lemmas für reguläre Sprachen wieder.

Beweisen Sie mit Hilfe des Pumping-Lemmas, dass die Sprache $L = \{0^n 1^n \,;\, n \geq 1\}$ nicht regulär ist.

Fortsetzung nächste Seite!

Aufgabe 2 (Grammatiken)

(a) Ein Palindrom ist eine Zeichenkette, die von vorn und von hinten gelesen gleich bleibt. Beispiele sind `otto` oder `madamimadam` ("Madam, I'm Adam").
Wir betrachten Palindrome über dem Alphabet $\{0, 1\}$. Geben Sie eine kontextfreie Grammatik G mit Terminalsymbolen, Nichtterminalsymbolen und Produktionen an, die alle Palindrome über $\{0, 1\}$ erzeugt. Begründen Sie, warum die von Ihnen angegebene Grammatik nur Palindrome erzeugt. Geben Sie die Ableitung des Palindroms 11011 mit Ihrer Grammatik an.

(b) Betrachten Sie die Grammatik $G = (\{E\}, \{0, 1, \ldots, 9\}, P, E)$ mit $P = \{E \to E{+}E \mid E - E \mid E \star E \mid E/E \mid 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9\}$ zur Beschreibung arithmetischer Ausdrücke mit den Operatoren $\{+, -, \star, /\}$ über den Ziffern $\{0, \ldots, 9\}$. Beweisen Sie, dass diese Grammatik mehrdeutig ist.

(c) Geben Sie eine eindeutige kontextfreie Grammatik zur Beschreibung arithmetischer Ausdrücke mit den Operatoren $\{+, -, \star, /\}$ über den Ziffern $\{0, \ldots, 9\}$ an, vgl. Aufgabenteil (b). Beschreiben Sie die Arbeitsweise der Produktionen. Begründen Sie, warum die von Ihnen angegebene Grammatik eindeutig ist.

(d) Die Sprache $L = \{a^n b^n c^n; n \geq 1\}$ kann nicht durch eine kontextfreie Grammatik beschrieben werden. Geben Sie eine kontextsensitive Grammatik zur Beschreibung von L an. Beschreiben Sie die Arbeitsweise Ihrer Grammatik. Leiten Sie das Wort $aaabbbccc \in L$ mit Hilfe Ihrer Grammatik ab.

Aufgabe 3 (kontextfreie Sprachen und Kellerautomaten)

(a) Betrachten Sie die kontextfreie Sprache $L = \{a^n 1 b^n; n \geq 1\}$ über dem Alphabet $\{a, b, 1\}$. Geben Sie eine kontextfreie Grammatik G mit Terminalsymbolen, Nichtterminalsymbolen und Produktionen an, die L erzeugt.

(b) Konstruieren Sie einen deterministischen Kellerautomaten $K = (Q, \Sigma, \Gamma, \delta, q_0, z_0, f)$, der L erkennt, d.h. $L = L(K)$. Geben Sie eine genaue Definition aller Elemente des Kellerautomaten mit einer mathematisch exakten Definition der Übergangsrelation δ. Erläutern Sie die Arbeitsweise des Kellerautomaten und begründen Sie, warum K alle Worte aus L erkennt.

(c) Erläutern Sie den Unterschied zwischen nichtdeterministischen und deterministischen Kellerautomaten durch Angabe der exakten Definitionen. Welche Unterschiede in den Verarbeitungsschritten gibt es?

(d) Kann die Sprache $L' = \{a^n b^n; n \geq 1\}$ durch einen deterministischen Kellerautomaten erkannt werden? Begründen Sie Ihre Antwort.

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2009

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr**	**46113**
Kennwort: _____	**2009**	
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 5

Bitte wenden!

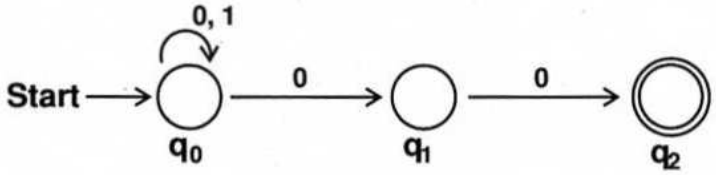
Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:

Zur Beschreibung regulärer Sprachen können deterministische endliche Automaten, nichtdeterministische endliche Automaten und reguläre Grammatiken eingesetzt werden.

a) Betrachten Sie folgenden nichtdeterministischen endlichen Automaten:



Erklären Sie die Funktionsweise dieses Automaten und die Rolle der verwendeten Zustände. Welche Sprache wird von dem Automaten erkannt?

b) Skizzieren Sie den Teilmengenkonstruktions-Mechanismus zur Berechnung eines äquivalenten deterministischen endlichen Automaten zu einem gegebenen nichtdeterministischen endlichen Automaten.
Erläutern Sie die Bedeutung der entstehenden Zustände des deterministischen endlichen Automaten.

c) Wandeln Sie den im Aufgabenteil a) angegebenen nichtdeterministischen endlichen Automaten mit Hilfe der Teilmengenkonstruktion in einen deterministischen endlichen Automaten um.
Geben Sie die Zwischenschritte des Verfahrens an und erläutern Sie die Berechnung der entstehenden Zustände.

d) Erläutern Sie den Mechanismus zur Konstruktion einer äquivalenten regulären Grammatik zu einem gegebenen deterministischen endlichen Automaten.
Erläutern Sie dabei insbesondere die Bedeutung der erzeugten Produktionen.

e) Konstruieren Sie zu dem in Aufgabenteil c) konstruierten deterministischen endlichen Automaten eine äquivalente reguläre Grammatik.
Geben Sie dabei insbesondere das Startsymbol, die Menge der verwendeten Nichtterminale sowie die Menge der erzeugten Produktionen an.
Erläutern Sie die Produktionen und deren Bedeutung.

f) Betrachten Sie folgende Sprache

$$L = \{x \in \{0,1\}^*;\ x \text{ stellt eine durch } 4, 8 \text{ oder } 16 \text{ teilbare Zahl im Dualsystem dar}\}$$

Ist diese Sprache regulär? Begründen Sie Ihre Antwort.

Fortsetzung nächste Seite!

Aufgabe 2:

Das Pumping-Lemma kann dazu verwendet werden, die Zugehörigkeit von Sprachen zu bestimmten Sprachklassen zu überprüfen.

a) Formulieren Sie die Aussage des Pumping-Lemmas für kontextfreie Sprachen.

b) Verwenden Sie das Pumping-Lemma für kontextfreie Sprachen um zu beweisen, dass die Sprache $L = \{0^n 1^n 2^n; n \geq 1\}$ nicht kontextfrei ist.
Erläutern Sie Ihre Vorgehensweise.

c) Betrachten Sie die beiden Sprachen:

$$L_1 = \{0^n 1^n 2^i; n \geq 1; i \geq 1\}$$

$$L_2 = \{0^i 1^n 2^n; n \geq 1; i \geq 1\}$$

Beweisen Sie, dass beide Sprachen kontextfrei sind.
Geben Sie jeweils zwei kontextfreie Grammatiken G_1 und G_2 an, die L_1, bzw. L_2 erzeugen.
Betrachten Sie die Sprache $L' = L_1 \cap L_2$, die den Schnitt von L_1 und L_2 beschreibt. Welche Worte enthält L'?
Beweisen oder widerlegen Sie die Aussage "Der Schnitt zweier kontextfreier Sprachen ist ebenfalls kontextfrei."

d) Konstruieren Sie einen nichtdeterministischen Kellerautomaten, der die Sprache $L_2 = \{0^i 1^n 2^n; n \geq 1; i \geq 1\}$ aus Aufgabenteil (c) erkennt.
Geben Sie eine genaue Definition aller Elemente des Kellerautomaten mit einer mathematisch exakten Definition der Übergangsrelation δ. Erläutern Sie die Arbeitsweise des von Ihnen konstruierten Kellerautomaten und begründen Sie, warum der Kellerautomat alle Worte aus L_2 akzeptiert.

e) Erläutern Sie den Unterschied zwischen nichtdeterministischen und deterministischen Kellerautomaten durch Angabe der exakten Definitionen. Welche Unterschiede in den Verarbeitungsschritten gibt es?

f) Kann die Sprache L_2 aus Aufgabenteil (d) durch einen deterministischen Kellerautomaten erkannt werden? Begründen Sie Ihre Antwort.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:
Für natürliche Zahlen a, b mit $b > 0$ bezeichne $\mathtt{div}(a, b)$ den Quotienten und $\mathtt{mod}(a, b)$ den Rest der ganzzahligen Division von a durch b, d.h. es gilt

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

und die Divisionseigenschaft der ganzen Zahlen besagt, dass $\mathtt{div}(a, b)$ und $\mathtt{mod}(a, b)$ durch diese Beziehung eindeutig bestimmt sind. Für $b = 0$ sind $\mathtt{div}(a, b)$ und $\mathtt{mod}(a, b)$ nicht definiert. Mit $\mathtt{ggT}(a, b)$ wird der größte gemeinsame Teiler von a und b bezeichnet, wobei $\mathtt{ggT}(a, 0) = a$ 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 Berechnung von `ggT` an.

Aufgabe 2:
Betrachten Sie über dem Alphabet $\Sigma = \{a, b\}$ die Sprache

$$L = \{w \in \Sigma^* \,;\, |w|_a \text{ ist ungerade und } |w|_b \text{ ist durch 3 teilbar}\},$$

wobei $|w|_a$ die Anzahl der a's in w und $|w|_b$ die Anzahl der b's in w ist. Konstruieren Sie den minimalen deterministischen endlichen Automaten, der die Sprache L akzeptiert.

Aufgabe 3:
Betrachten Sie über dem Alphabet $\Sigma = \{a, b\}$ die Sprache

$$L = \{a^r b^s \,;\, r \leq s \leq 2r\}.$$

a) Beweisen Sie, dass die Sprache L nicht regulär ist.

b) Konstruieren Sie einen Kellerautomaten, der die Sprache L akzeptiert.

Fortsetzung nächste Seite!

Aufgabe 4:

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.

Aufgabe 5:

Sei Σ ein endliches Alphabet.

a) Beschreiben Sie einen Algorithmus, der für beliebige endliche Automaten $\mathcal{A} = (Q_A, \Sigma, \delta_A, q_A, F_A)$ und $\mathcal{B} = (Q_B, \Sigma, \delta_B, q_B, F_B)$ entscheidet, ob sie die gleiche Sprache akzeptieren, d.h. ob $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$.

b) Zeigen Sie, dass es keinen Algorithmus gibt, der für beliebige deterministische Turingmaschinen $\mathcal{A} = (Q_A, \Sigma, \Gamma_A, \delta_A, q_A, F_A)$ und $\mathcal{B} = (Q_B, \Sigma, \Gamma_B, \delta_B, q_B, F_B)$ entscheidet, ob sie die gleiche Sprache akzeptieren, d.h. ob $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$.

46113

Theoretische Informatik (nicht vertieft)

Herbst 2009

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2009**	**46113**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **5**

Bitte wenden!

Thema Nr. 1

Aufgabe 1: Reguläre Mengen

Sei $L \subseteq \{a,b\}^*$ die Menge aller Worte, deren zweites und zweitletztes Zeichen verschieden sind. Das leere Wort gehört nicht zu L.

(a) Konstruieren Sie einen regulären Ausdruck für L.

(b) Konstruieren Sie einen deterministischen endlichen Automaten A mit $L(A) = L$.

Aufgabe 2: Konstruktion von Turingmaschinen

a) Konstruieren Sie eine (k-Band) Turingmaschine M für die Sprache
$PAL = \{wcw^R \mid w \in \{a,b\}^*\}$. Dabei ist w^R die Spiegelung von w.

b) Geben Sie die Menge der Worte der Länge 5 und 6 an, die Element von L sind.

c) Welche Zeit-Komplexität hat Ihre Turingmaschine M?

d) Ordnen Sie PAL in die folgende Hierarchie ein:
die regulären Mengen, die kontextfreien Sprachen, P, NP, die rekursiven (entscheidbaren) Mengen.
Was ist die kleinste Klasse in der PAL liegt?

Aufgabe 3: Formale Sprachen

a) Zeigen Sie, dass die Sprache $L = \{a^n b^m c^q \mid n,m,q \geq 0,\ n=m \text{ oder } n = q\}$ kontextfrei ist.

Es reicht die Angabe einer geeigneten Grammatik oder eines geeigneten Automaten.

b) Zeigen Sie, dass die (obige) Sprache $L = \{a^n b^m c^q \mid n,m,q \geq 0,\ n=m \text{ oder } n = q\}$ nicht regulär ist.

Der Beweis z. B. des Pumping Lemmas und von Abschlusseigenschaften wird nicht verlangt.

Fortsetzung nächste Seite!

Aufgabe 4: Abschlusseigenschaften

a) Zeigen Sie: Wenn L eine reguläre Sprache ist, dann ist auch L* regulär.

Es reicht die Angabe der Konstruktion; ein Korrektheitsbeweis Ihrer Konstruktion ist nicht verlangt.

b) Zeigen Sie: Wenn L eine kontextfreie Sprache ist, dann ist auch L* kontextfrei.

Es reicht die Angabe der Konstruktion; ein Korrektheitsbeweis Ihrer Konstruktion ist nicht verlangt.

Aufgabe 5: Komplexität

Begründen Sie, warum $NP \subseteq REK$ gilt.
NP ist die Klasse der nicht-deterministisch in Polynomzeit lösbaren Probleme (Sprachen);
REK ist die Klasse der entscheidbaren (oder rekursiven) Sprachen.

Thema Nr. 2

Aufgabe 1: Reguläre Sprachen

Reguläre Sprachen können durch reguläre Ausdrücke mit den Operatoren $\bullet$ (Konkatenation), $+$ (Vereinigung) und $\star$ (Abschluss) definiert werden.

(a) Geben Sie eine genaue induktive Definition von regulären Ausdrücken über einem gegebenen Alphabet $\sum$ mit Basis- und Induktionsteil.

(b) Geben Sie reguläre Ausdrücke für die folgenden regulären Mengen an:

(α) die Menge aller Binärzahlen über $\{0,1\}^\star$, die durch 4 teilbar sind,

(β) die Menge aller Worte über $\{0,1\}^\star$, die abwechselnd 0 und 1 enthalten,

(γ) die Menge aller Worte über $\{0,1\}^\star$, die eine gerade Anzahl von 0en und 1en enthalten.

(c) Neben regulären Ausdrücken können auch endliche Automaten zur Beschreibung regulärer Mengen verwendet werden. Dabei kann zu jedem beliebigen regulären Ausdruck R über einem Alphabet $\sum$ ein endlicher Automat konstruiert werden, der die von R definierte Sprache $L(R)$ erkennt. Zeigen Sie, wie diese Konstruktion für einen beliebigen regulären Ausdruck R funktioniert. Geben Sie dabei entsprechend der induktiven Definition aus Aufgabenteil (a) für jeden Teil der Basis und des Induktionsschritts den zu den definierten regulären Ausdrücken gehörenden endlichen Automaten an. Begründen Sie Ihre Konstruktionsschritte. Ist der so entstehende Automat deterministisch oder nicht-deterministisch?

(d) Verwenden Sie die Konstruktionsvorschrift aus Aufgabenteil (c) zur schrittweisen Konstruktion eines Automaten zum regulären Ausdruck $\mathbf{1(0+1)}^\star$ über dem Alphabet $\{0,1\}$. Begründen Sie die durchgeführten Schritte.

Fortsetzung nächste Seite!

Aufgabe 2: Kontextfreie Sprachen und Grammatiken

(a) Betrachten Sie die folgende Sprache $L_1 = \{a^n b^m c^m d^n \text{ für } m \geq 0, n \geq 0\}$. Ist dies eine kontextfreie Sprache? Begründen Sie Ihre Antwort. Wenn L_1 eine kontextfreie Sprache ist, geben Sie eine kontextfreie Grammatik mit Terminalsymbolen, Nichtterminalsymbolen und Produktionen an, die L_1 erzeugt. Erläutern Sie die Arbeitsweise der von Ihnen angegebenen Grammatik.

(b) Formulieren Sie die Aussage des Pumping-Lemmas für kontextfreie Sprachen. Erläutern Sie die Gültigkeit dieses Lemmas. Erläutern Sie, wie das Pumping-Lemma für kontextfreie Sprachen dazu verwendet werden kann zu beweisen, dass eine bestimmte Sprache nicht kontextfrei ist.

(c) Betrachten Sie die folgende Sprache $L_2 = \{a^n b^m c^n d^m \text{ für } m \geq 0, n \geq 0\}$. Beweisen Sie mit Hilfe des Pumping-Lemmas für kontextfreie Sprachen, dass L_2 nicht kontextfrei ist.

(d) Argumentieren Sie, warum die kontextfreien Sprachen unter der Vereinigungsoperation abgeschlossen sind. Konstruieren Sie dazu für beliebige kontextfreie Grammatiken G_1 bzw. G_2 zur Erzeugung von kontextfreien Sprachen L_1 bzw. L_2 eine kontextfreie Grammatik zur Erzeugung von $L_1 \cup L_2$. Erläutern Sie die Arbeitsweise von G.

Aufgabe 3: Kellerautomaten

(a) Kontextfreie Sprachen können mit Hilfe von Kellerautomaten erkannt werden. Geben Sie eine exakte mathematische Definition eines Kellerautomaten. Erläutern Sie den Unterschied zwischen nicht-deterministischen und deterministischen Kellerautomaten. Welche Unterschiede in den Verarbeitungsschritten gibt es?

(b) Betrachten Sie die Sprache $L_1 = \{a^n b^m c^m d^n \text{ für } m \geq 0, n \geq 0\}$ aus Aufgabe 2, Teil (a). Konstruieren Sie einen (nicht-deterministischen) Kellerautomaten $K = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, der L_1 erkennt. Geben Sie eine genaue Definition aller Elemente des Kellerautomaten mit einer mathematisch exakten Definition der Übergangsrelation δ an. Erläutern Sie die Arbeitsweise des Kellerautomaten und begründen Sie, warum K alle Worte aus L erkennt.

(c) Kann die Sprache aus Aufgabenteil (b) durch einen deterministischen Kellerautomaten erkannt werden? Begründen Sie Ihre Antwort.

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2010

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr**	**46113**
Kennwort: _____	**2010**	
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

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!

1. Betrachten Sie über dem Alphabet $\Sigma = \{a, b\}$ die Sprache L, die durch folgendes Regelsystem vollständig gegeben ist:

- Das Wort a gehört zu L.
- Gehört ein Wort w zu L und endet w mit einem a, so gehört auch wb zu L, d.h. für $u \in \Sigma^*$ gilt $ua \in L \Rightarrow uab \in L$.
- Gehört w zu L, so gilt auch $ww \in L$.
- Folgen in einem Wort $w \in L$ drei a's unmittelbar aufeinander, so gehört auch das Wort zu L, das man aus w erhält, wenn man aaa durch b ersetzt, d.h. für $u, v \in \Sigma^*$ gilt $uaaav \in L \Rightarrow ubv \in L$.
- Folgen in einem Wort $w \in L$ zwei b's unmittelbar aufeinander, so gehört auch das Wort zu L, das man aus w erhält, wenn man bb streicht, d.h. für $u, v \in \Sigma^*$ gilt $ubbv \in L \Rightarrow uv \in L$.

Beispielsweise gehört das Wort $baab$ zu L, was man an der Ableitung

$$a \vdash aa \vdash aaaa \vdash ba \vdash bab \vdash babbab \vdash baab$$

sieht. Da das Regelsystem sowohl verlängernde als auch verkürzende Regeln enthält, ist die Zugehörigkeit eines Worte $w \in \Sigma^*$ zur Spache L scheinbar nicht einfach zu entscheiden. Da das Regelsystem keine reguläre Grammatik ist, ist auch nicht klar, ob die Sprache L regulär ist.

a) Zeigen Sie, dass alle Wörter $w \in L$ die Eigenschaft haben:

(*) Die Anzahl $|w|_a$ der Vorkommen von a in w ist nicht durch 3 teilbar.

b) Tatsächlich charakterisiert die Eigenschaft (*) die Sprache L, d.h. es gilt

$L = \{w \in \Sigma^* \,;\, 3 \nmid |w|_a\}$.

Verwenden Sie diese Aussage (die Sie nicht beweisen sollen!), um einen minimalen vollständigen DFA zu konstruieren, der die Sprache L akzeptiert.

c) Beweisen Sie, dass Ihr Automat tatsächlich minimal ist!

2. Betrachten Sie die Funktion des ganzzahligen Quadratwurzelziehens:

$$\texttt{sqrt} : \mathbb{N} \to \mathbb{N} : x \mapsto \lfloor \sqrt{x} \rfloor$$

a) Geben Sie ein Programm für `sqrt` in einer Ihnen geläufigen Programmiersprache an.

b) Geben Sie ein WHILE-Programm für `sqrt` an.

c) Geben Sie ein LOOP-Programm für `sqrt` an.

d) Zeigen Sie, dass `sqrt` eine primitiv-rekursive Funktion ist, indem Sie eine entsprechende Definition angeben.

Fortsetzung nächste Seite!

Hinweise:

- Sie können, falls Ihnen das hilfreich erscheint, die Tatsache verwenden, dass $1 + 3 + 5 + \cdots + (2n - 1) = n^2$ ist.
- $\lfloor x \rfloor$ bezeichnet die größte ganze Zahl, die kleiner oder gleich x ist, also z.B. $\lfloor 3,1 \rfloor = 3$.

3. Beweisen oder widerlegen Sie die folgenden Behauptungen über Sprachen $A, B \subseteq \Sigma^*$. $A \Delta B = (A \setminus B) \cup (B \setminus A)$ bezeichne dabei die symmetrische Mengendifferenz.

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

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

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

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

e) Sind A und B polynomiell-entscheidbar, so ist auch $A \Delta B$ polynomiell-entscheidbar.

4. ALICE und BOB verfügen jeder über einen DFA über dem Alphabet $\Sigma = \{0, 1\}$. Der Automat von ALICE sei $\mathfrak{A}$, der Automat von BOB sei $\mathfrak{B}$. Beide Automaten haben die gleiche Anzahl n von Zuständen. ALICE und BOB wollen die Frage klären, ob es ein Wort $w \in \Sigma^*$ gibt, das von beiden Automaten akzeptiert wird.

> ALICE argumentiert: "Diese Frage können wir gar nicht entscheiden, denn dafür müssten wir unendlich viele Wörter aus Σ^* durchprobieren."
>
> BOB hält dem entgegen: "Das geht doch ganz einfach: Wenn ein Automat n Zustände hat, können alle akzeptierenden Zustände in höchstens $n - 1$ Schritten erreicht werden. Es genügt also, Wörter der Länge $< n$ zu testen, ob eines von ihnen von beiden Automaten $\mathfrak{A}$ und $\mathfrak{B}$ akzeptiert wird."

Kommentieren Sie die beiden Aussagen! Sollten Sie weder ALICE noch BOB zustimmen, so zeigen Sie einen Weg auf, die Frage zu beantworten. Geben Sie ein Entscheidungsverfahren an oder weisen Sie nach, dass es sich um ein unentscheidbares Problem handelt.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

1. Gegeben ist die folgende Sprache $L1$ über dem Alphabet $\Sigma = \{a, b\}$:

$L1 = \{w \in \Sigma^* \mid$ die Anzahl der a in w ist gerade und b kommt in w genau einmal vor$\}$.

a) Geben Sie einen deterministischen endlichen Automaten an, der die Sprache $L1$ akzeptiert.

b) Geben Sie einen regulären Ausdruck an, der die Sprache $L1$ beschreibt.

Die folgende Sprache $L2$ ist eine Erweiterung von $L1$:
$L2 = \{w \in \Sigma^* \mid$ die Anzahl der a in w ist gerade und die Anzahl der b in w ist ungerade$\}$.

c) Geben Sie einen deterministischen endlichen Automaten an, der die Sprache $L2$ akzeptiert.

d) Geben Sie eine rechtslineare Grammatik an, die die Sprache $L2$ erzeugt.

2. Gegeben ist die folgende Sprache L über dem Alphabet $\Sigma = \{0, 1\}$:

$L = \{w \in \Sigma^* \mid w \neq \epsilon, w$ enthält gleich viele 0-en wie 1-en und in jedem Anfangswort von w kommen mindestens soviele 0-en wie 1-en vor$\}$.

a) Geben Sie alle Wörter $w \in L$ mit Länge $|w| \leq 4$ an.

b) Geben Sie eine kontextfreie Grammatik an, die die Sprache L erzeugt.

c) Zeigen Sie, dass L nicht regulär ist. Der Beweis kann ohne Anwendung des Pumping Lemmas unter Verwendung der beiden folgenden Tatsachen geführt werden:

i) Der Durchschnitt zweier regulärer Sprachen ist regulär.

ii) Die Sprache $L' = \{0^j 1^j \mid j > 0\}$ ist nicht regulär.

Wir betrachten nun für alle natürlichen Zahlen $n \in \mathbb{N}$ die folgenden Teilsprachen L_n von L:
$L_n = \{w \in L \mid$ in jedem Anfangswort von w kommen höchstens n mehr 0-en vor als 1-en$\}$.

d) Für alle natürlichen Zahlen n ist L_n regulär. Zeigen Sie dies für den Fall $n = 3$.

e) Wie kann man aus den Tatsachen, dass L nicht regulär ist und L_n für alle $n \in \mathbb{N}$ regulär ist, schließen, dass die Vereinigung abzählbar unendlich vieler regulärer Mengen nicht notwendigerweise regulär ist?

3. Zeigen Sie, dass die Funktion $f : \mathbb{N} \to \mathbb{N}$, mit $f(n) = 2^{(n \; div \; 2)}$ für alle $n \in \mathbb{N}$, primitiv rekursiv ist. Dabei bezeichnet $n \; div \; 2$ die ganzzahlige Division durch 2.

Beim Beweis ist das Schema der primitiven Rekursion zu verwenden, d.h. es sind geeignete primitiv rekursive Funktionen $g : \mathbb{N} \to \mathbb{N}$ und $h : \mathbb{N}^2 \to \mathbb{N}$ anzugeben, so dass gilt:
$f(0) = g, f(n+1) = h(n, f(n))$ für alle $n \in \mathbb{N}$.

Es kann vorausgesetzt werden, dass die Multiplikation und die Fallunterscheidung, ob eine natürliche Zahl gerade ist oder nicht, primitiv rekursiv sind.

Fortsetzung nächste Seite!

4. Entscheiden Sie, welche der folgenden Aussagen richtig und welche falsch sind und geben Sie jeweils eine kurze Begründung für Ihre Antwort an. Unter einer totalen Funktion wird im Folgenden, wie üblich, eine auf allen Argumenten definierte Funktion verstanden.

a) Jede μ-rekursive totale Funktion ist primitiv rekursiv.

b) Jede primitiv rekursive Funktion ist eine totale μ-rekursive Funktion.

c) Das Komplement jeder primitiv rekursiven Teilmenge M der natürlichen Zahlen ist primitiv rekursiv. (Eine Menge heißt primitiv rekursiv, wenn ihre charakteristische Funktion primitiv rekursiv ist.)

d) Das Komplement jeder entscheidbaren Teilmenge M der natürlichen Zahlen ist entscheidbar.

e) Das Komplement jeder rekursiv aufzählbaren Teilmenge M der natürlichen Zahlen ist rekursiv aufzählbar.

46113

Theoretische Informatik (nicht vertieft)

Herbst 2010

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2010**	**46113**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **5**

Bitte wenden!

Thema Nr. 1

Aufgabe 1 **Chomsky-Hierarchie**

a) Zeigen Sie jeweils durch Angabe einer Sprache, dass die Typ 3-Sprachen echt in der Menge der Typ 2-Sprachen, sowie die Typ 2-Sprachen echt in der Menge der Typ 1-Sprachen enthalten sind.

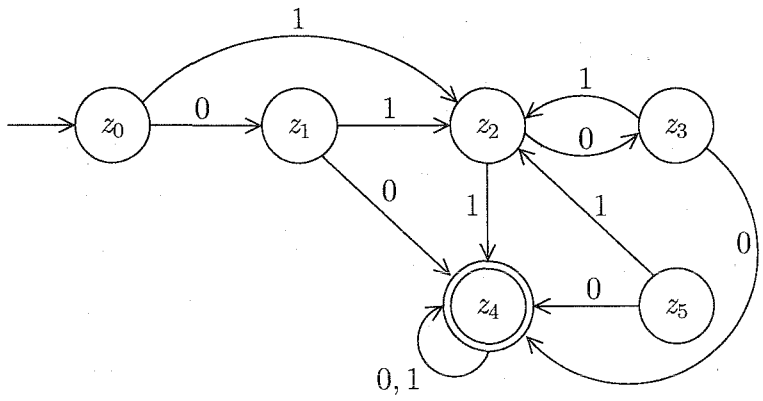
b) Gegeben sei die Sprache

$$L = \{x \in \{0,1,2,3\}^* \setminus \{\varepsilon\} \mid x \text{ (als Zahl im Dezimalsystem gelesen) ist } \textit{nicht} \text{ durch 3 teilbar}\}.$$

Ordnen Sie die Sprache L in die Chomsky-Hierarchie ein und beweisen Sie Ihre Aussage durch Angabe eines akzeptierenden Automaten.

Aufgabe 2 **Endliche Automaten**

Gegeben sei der folgende deterministische, endliche Automat D:



a) Konstruieren Sie aus D einen Minimalautomaten D_{min}.

b) Geben Sie explizit die Sprache an, die der Automat D bzw. D_{min} akzeptiert.

c) Geben Sie einen regulären Ausdruck für diese Sprache an.

Aufgabe 3 **Pumping Lemma und Satz von Myhill-Nerode**

Sei $\Sigma = \{a, b, c\}$ ein Alphabet und seien die folgenden beiden kontextfreien, nicht-regulären Sprachen über Σ gegeben:

$$L_1 = \{a^m b^n c^n \mid m \geq 0, n \geq 1\}$$
$$L_2 = L_1 \cup \{b, c\}^*$$

a) Geben Sie eine kontextfreie Grammatik für L_1 an.

b) Zeigen Sie, dass L_2 die Behauptung des Pumping Lemma für reguläre Sprachen erfüllt.

c) Geben Sie den Satz von Myhill-Nerode an.

d) L_2 ist eine nicht-reguläre Sprache. Dies kann nach Teil b) dieser Aufgabe nicht mit dem Pumping Lemma gezeigt werden. Beweisen Sie nun mit Hilfe des Satzes von Myhill-Nerode, dass L_2 nicht-regulär ist. [Hinweis: Es reicht, unendlich viele nicht-äquivalente Wörter $z \in \Sigma^*$ zu finden.]

Fortsetzung nächste Seite!

Aufgabe 4 **Berechenbarkeit**

Sei eine Funktion f gegeben, die für jede Eingabe $w \in \{0,1\}^*$ das Wort $w' \in \{0,1\}^*$ berechnet, das aus w entsteht, indem man jedes Vorkommen von 0 durch 00 ersetzt.

Beispiel: $f(00101) = 00001001$

a) Geben Sie eine deterministische Turing-Maschine M an, die f berechnet. Beschreiben Sie dabei die Bedeutung der Zustände bzw. Übergänge Ihrer Turing-Maschine informell.

b) Geben Sie den Ablauf von M für die Eingabe 01 an.

c) Bestimmen Sie Laufzeit- und Speicherplatzkomplexität von M in O-Notation.

Thema Nr. 2

Annahmen:
Sie dürfen als bekannt und bewiesen voraussetzen:
Die Sprache $\{a^n b^n \mid n \geq 1\}$ ist nicht regulär.
Die Sprache $\{a^n b^n c^n \mid n \geq 1\}$ ist nicht kontextfrei.
Um zu zeigen, dass eine Sprache L regulär (kontextfrei) ist, reicht die Angabe einer entsprechenden Beschreibung (Automat, Grammatik, regulärer Ausdruck). Sie müssen nicht mehr zeigen, dass Ihre Beschreibung korrekt ist und genau die vorgegebene Sprache beschreibt.

Aufgabe 1: reguläre Mengen

Sei L die Menge aller Worte über dem Alphabet {a,b}, bei denen das zweite und das zweitletzte Zeichen gleich sind.
Beachten Sie auch die kurzen Worte.

Beschreiben Sie L
a) durch einen regulären Ausdruck
b) durch einen deterministischen endlichen Automaten A.

Aufgabe 2: regulär oder nicht

L besteht aus der Menge aller Worte über dem Alphabet {a,b}, bei denen die Anzahl von a's gerade und die Anzahl der b's ungerade ist.

Ist die Sprache L regulär oder nicht?
Begründen Sie Ihre Antwort.

Aufgabe 3:

Sei $PAL = \{w \mid w = w^{rev},\ w \in \{a,b\}^*\}$, w^{rev} ist w gespiegelt oder rückwärts gelesen.

a) Geben Sie alle Worte bis zur Länge 5 von PAL an.

b) Klassifizieren Sie PAL:
Ist PAL regulär? Begründen Sie Ihre Entscheidung.
Ist PAL kontextfrei? Begründen Sie Ihre Entscheidung.
Ist PAL rekursiv aufzählbar (positiv semi-entscheidbar)? Begründen Sie Ihre Entscheidung.

Fortsetzung nächste Seite!

Aufgabe 4:

Konstruieren Sie eine Turingmaschine M für die Sprache
$L = \{a^n b^n c^n \mid n \geq 1\}$

Welche Zeit-Komplexität hat Ihre Turingmaschine M?
Welche Speicher-Komplexität hat Ihre Turingmaschine M?

Aufgabe 5:

Das Feedback-Arc-Set Problem (FAS)
Gegeben: Ein gerichteter Graph $G = (V, E)$ mit Knotenmenge V und Kantenmenge E und eine Zahl k.
Problem: Gibt es eine Teilmenge von höchstens k Kanten F, $F \subseteq E$, $|F| \leq k$,
so dass der Graph $G' = (V, E \backslash F)$ keinen Zyklus mehr enthält?
Warum ist dieses Problem in NP? Erläutern Sie dies.

Hinweis: Es ist **nicht** gefragt, dass FAS NP-hart ist. Das hat R. Karp 1970 gezeigt.

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2011

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

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

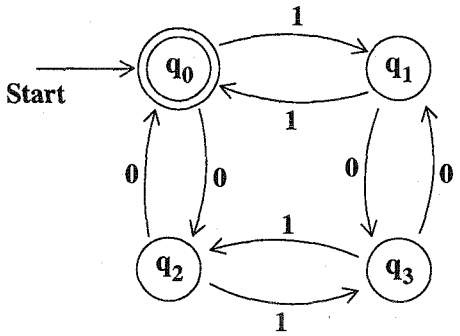
Bitte wenden!

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

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

Modifizieren Sie den Automaten so, dass er die Sprache

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

erkennt. Begründen Sie die von Ihnen vorgeschlagene Modifikation.

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

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

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 $\boldsymbol{p} = p_1 p_2 \ldots p_n$ 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 $\Sigma_k = \{0, \pm 1, \pm 2, \ldots, \pm k\}$ 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 $\boldsymbol{p} = p_1 p_2 \ldots p_n$ versteht man das Wort $\delta(\boldsymbol{p}) = w_1 w_2 \ldots w_{n-1} \in \Sigma_1^*$ 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:

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

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.

46113

Theoretische Informatik (nicht vertieft)

Herbst 2011

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2011**	**46113**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **6**

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:
reguläre Sprachen

a) 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?

b) Beweisen Sie mit Hilfe des Pumping-Lemmas für reguläre Sprachen, dass die Sprache $L_1 = \{a^n b^n a^m b^m \mid m \geq 1, n \geq 1\}$ nicht regulär ist. Erläutern Sie Ihre Argumentation.

c) Sei L eine reguläre Sprache. Sei n die Zahl des Pumping-Lemmas zu L. Beweisen Sie, dass folgende Aussage gilt:
$\mid L \mid = \infty \Leftrightarrow$ es gibt in L ein Wort w mit $n \leq \mid w \mid < 2n$

d) Begründen Sie, warum reguläre Sprachen unter Vereinigung abgeschlossen sind. Betrachten Sie dazu zwei reguläre Sprachen L und M und begründen Sie, warum auch $L \cup M$ eine reguläre Sprache ist.

e) Sei $A = (Q, \Sigma, \delta, q_0, F)$ ein deterministischer endlicher Automat, der eine reguläre Sprache L erkennt. Konstruieren Sie zu A einen deterministischen endlichen Automaten B, der die reguläre Sprache $\bar{L} = \Sigma^* - L$ akzeptiert. Erläutern Sie die Funktionsweise des von Ihnen konstruierten Automaten.

Fortsetzung nächste Seite!

Aufgabe 2:
kontextfreie Sprachen und Kellerautomaten

a) Betrachten Sie die folgende Sprache

$$L_1 = \{a^n b^n a^m b^m \mid m \geq 1, n \geq 1\}$$

Geben Sie eine kontextfreie Grammatik mit Terminalsymbolen, Nichtterminalsymbolen und Produktionen an, die L_1 erzeugt. Erläutern Sie die Arbeitsweise der von Ihnen angegebenen Grammatik.

b) Definieren Sie die Begriffe Linksableitung und Rechtsableitung einer kontextfreien Grammatik. Geben Sie für die von Ihnen in Aufgabenteil (a) konstruierte Grammatik eine Linksableitung für das Wort $abaabb \in L_1$ an.

c) Kontextfreie Sprachen können mit Hilfe von Kellerautomaten erkannt werden. Geben Sie eine mathematisch exakte Definition eines Kellerautomaten an. Erläutern Sie den Unterschied zwischen nicht-deterministischen und deterministischen Kellerautomaten. Welche Unterschiede in den Verarbeitungsschritten gibt es?

d) Konstruieren Sie einen Kellerautomaten $K = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, der die Sprache L_1 aus Aufgabenteil (a) akzeptiert. Geben Sie eine mathematisch exakte Definition der Übergangsrelation δ an. Erläutern Sie die Arbeitsweise des von Ihnen konstruierten Kellerautomaten K und begründen Sie, warum K alle Worte aus L_1 akzeptiert. Ist der von Ihnen konstruierte Kellerautomat deterministisch oder nicht-deterministisch? Begründen Sie, warum L_1 durch einen deterministischen Kellerautomaten erkannt werden kann.

e) Formulieren Sie die Aussage des Pumping-Lemmas für kontextfreie Sprachen. Erläutern Sie die Gültigkeit dieses Lemmas. Erläutern Sie, wie das Pumping-Lemma für kontextfreie Sprachen dazu verwendet werden kann zu beweisen, dass eine gegebene Sprache nicht kontextfrei ist.

f) Betrachten Sie die folgende Sprache $L_2 = \{a^n b^m a^n b^m \mid m \geq 1, n \geq 1\}$. Beweisen Sie mit Hilfe des Pumping-Lemmas für kontextfreie Sprachen, dass L_2 nicht kontextfrei ist.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Annahmen:

Sie dürfen als bekannt und bewiesen voraussetzen:
Die Sprache $\{a^n b^n c^n \mid n \geq 1\}$ ist nicht kontext-frei.

Um zu zeigen, dass eine Sprache L regulär (bzw. konxtexfrei) ist, reicht die Angabe einer entsprechenden Beschreibung (Automat, Grammatik, Ausdruck). Sie müssen nicht mehr zeigen, dass Ihre Beschreibung korrekt ist und genau die vorgegebene Sprache beschreibt.

Aufgabe 1: reguläre Mengen

Sei L die Menge aller Worte über dem Alphabet $\{0,1\}$, die eine durch vier teilbare Binärzahl ohne führende Nullen darstellen.

a) Beschreiben Sie L durch einen regulären Ausdruck.

b) Geben Sie eine vollständige Beschreibung eines deterministischen endlichen Automaten A für L an.

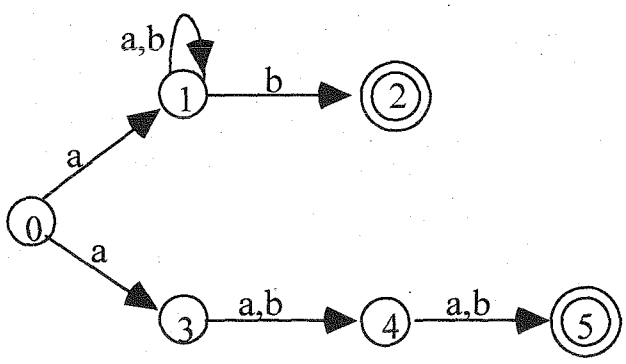
c) Wenn Ihr Automat A nicht minimal ist, konstruieren Sie einen äquivalenten minimalen Automaten.

d) Beschreiben und begründen Sie in Worten, warum der Automat aus Teilaufgabe 1 b bzw. 1 c minimal ist.
Durch welche Eigenschaft(en) wird ein minimaler Automat charakterisiert?

Fortsetzung nächste Seite!

Aufgabe 2: deterministisch

Konstruieren Sie zum folgenden Automaten einen äquivalenten deterministischen endlichen Automaten. Anfangszustand ist der Zustand 0, akzeptierende Endzustände sind 2 und 5:



Aufgabe 3: regulär oder nicht

$L = \{w \in \{a,b,c\}^* \mid w = a^i\ b^j\ c^k \text{ mit } i,j,k \geq 0 \text{ und } j=k\}$

Ist die Sprache L regulär oder nicht?
Begründen Sie Ihre Antwort.

Aufgabe 4:

Zeigen Sie, dass die Sprache

$L = \{w \in \{a,b,c\}^* \mid w = a^i\ b^j\ c^k \text{ mit } i,j,k \geq 1 \text{ und } (j=k \text{ oder } i=j)\}$

kontextfrei ist.
Es reicht die Angabe einer entsprechenden Grammatik bzw. eines Kellerautomaten.

Fortsetzung nächste Seite!

Aufgabe 5:

a) Konstruieren Sie eine Turingmaschine für die Sprache
$L = \{ u \# v \mid u,v \in \{a,b\}^* \text{ und } u \neq v \}$.

b) Welche Zeit-Komplexität hat Ihre Turingmaschine M?
Welche Speicher-Komplexität hat Ihre Turingmaschine M?

Aufgabe 6: NP

Sei $BE = \{\alpha \mid \alpha$ ist ein erfüllbarer boolescher Ausdruck über $\wedge, \vee, \neg, \Rightarrow, \Leftarrow, \Leftrightarrow\}$

Begründen Sie in Worten:
a) BE ist in **NP.**

b) BE ist NP-vollständig.
Hierzu darf nach dem Satz von Cook & Levin vorausgesetzt werden, dass das Erfüllbarkeitsproblem (SAT) NP-vollständig ist.

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2012

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr 2012**	**46113**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **4**

Bitte wenden!

Thema Nr. 1

1. **Binärzahlen mit Prüfziffer**

   Sei $\Sigma = \{0, 1\}$. Eine *Binärzahl mit Prüfziffer* ist ein Wort der Sprache

$$\mathcal{L}_p = \{b_0 \dots b_n b_c \mid n \geq 0 \wedge b_c = \begin{cases} 0, & \text{falls die Anzahl der 1en in } b_0 \dots b_n \\ & \text{gerade ist} \\ 1, & \text{sonst} \end{cases} \}.$$

   (a) Zeigen Sie durch Angabe einer Grammatik, dass $\mathcal{L}_p$ regulär ist.

   (b) Geben Sie die Äquivalenzklassen der Sprache $\mathcal{L}_p$ (bzgl. des Satzes von Myhill/Nerode) an.

   (c) Zeigen Sie mit dem Pumping-Lemma: Die Sprache

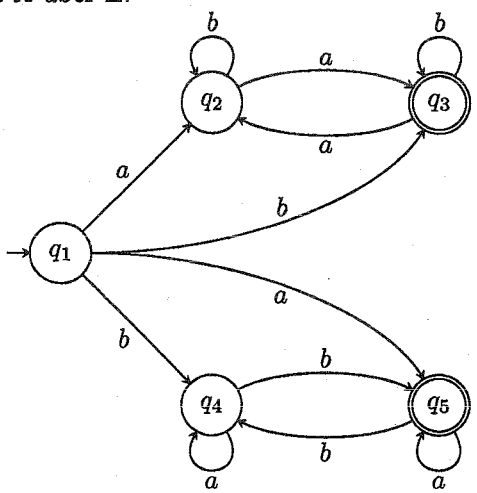
$$\mathcal{L}_{P'} = \{b_0 \dots b_n b_c \mid n \geq 0 \wedge b_c = \begin{cases} 0, & \text{falls die Anzahl der 1en in } b_0 \dots b_n \\ & \text{kleiner als die der 0en ist} \\ 1, & \text{sonst} \end{cases} \}$$

   ist nicht regulär.

   (d) Zeigen Sie: Die Sprache $\mathcal{L}_{P'}$ ist kontextfrei.

2. **Endliche Automaten**

   (a) Sei $\Sigma = \{a, b\}$. Gegeben sei der folgende nichtdeterministische endliche Automat A über Σ:



   Geben Sie die Sprache $\mathcal{L}(A)$ des Automaten an.

   (b) Konstruieren Sie mittels der Potenzmengenkonstruktion einen deterministischen endlichen Automaten A' mit $\mathcal{L}(A') = \mathcal{L}(A)$.

   (c) Minimieren Sie den Automaten A' oder zeigen Sie, dass dieser bereits minimal ist.

Thema Nr. 2

Annahmen:
Sie dürfen als bekannt und bewiesen voraussetzen:
Die Sprache $\{a^n b^n \mid n \geq 1\}$ ist nicht regulär.
Die Sprache $\{a^n b^n c^n \mid n \geq 1\}$ ist nicht kontextfrei.
Um zu zeigen, dass eine Sprache L regulär (kontextfrei) ist, reicht die Angabe einer entsprechenden Beschreibung (Automat, Grammatik, Ausdruck).
Sie müssen nicht mehr zeigen, dass Ihre Beschreibung korrekt ist und genau die vorgegebene Sprache beschreibt.

Aufgabe 1: reguläre Mengen

Sei L die Menge aller Worte über dem Alphabet {a, b}, bei denen das erste, dritte und das zweitletzte Zeichen gleich sind.

Beschreiben Sie L
a) durch einen regulären Ausdruck.
b) durch einen deterministischen endlichen Automaten A.

Beachten Sie auch die kurzen Worte!

Aufgabe 2: regulär oder nicht

L besteht aus der Menge aller Binärzahlen ohne führende Nullen über dem Alphabet {0,1}, die durch 3 oder durch 8 teilbar sind.

Ist die Sprache L regulär oder nicht?
Begründen Sie Ihre Antwort durch die Angabe einer passenden Beschreibung für L oder den Nachweis, dass L nicht regulär sein kann.

Aufgabe 3: Abschlusseigenschaften

Für zwei Sprachen X und Y sei $X \otimes Y = \{w \mid (w \in X \text{ und } w \in Y) \text{ oder } (w \notin X \text{ und } w \notin Y)\}$.

Begründen Sie, dass $X \otimes Y$ regulär ist, wenn X und Y regulär sind.

Fortsetzung nächste Seite!

Aufgabe 4: Minimierung von deterministischen endlichen Automaten

Minimieren Sie den DFA $A = (Q, \{a,b\}, \delta, 0, F)$ mit $Q = \{0,1,2,3,4,5\}$, $F = \{1,2,3\}$ und δ mit nachfolgender Tabelle

	0	1	2	3	4	5
a	1	2	3	1	4	5
b	4	3	1	2	5	4

Aufgabe 5:

Gegeben sei die Sprache L mit der Menge aller Worte über $\{a\}^* \{b\}^* \{a\}^*$, bei denen die Anzahl der a's gleich der Anzahl der b's ist.

a) Zeigen Sie, dass L kontextfrei ist.
b) Zeigen Sie, dass L nicht regulär ist.

Aufgabe 6:

a) Konstruieren Sie eine Turingmaschine M für die Sprache
$L = \{w\, c\, w^R\ c\ w \mid w \in \{a, b\}^+\}$
Dabei ist w^R die Spiegelung von w (w wird rückwärts gelesen).

b) Welche Zeit-Komplexität hat Ihre Turingmaschine M?

Aufgabe 7:

a) Definieren/Beschreiben Sie, wann ein Problem (eine Sprache) NP-vollständig ist.

b) Beschreiben Sie ein NP-vollständiges Problem.
Die Beschreibung kann etwa so aussehen:
Das Kürzeste-Wege-Problem:
Gegeben ist ein Graph G mit Längen auf den Kanten, einem Startknoten s und einem Zielknoten t und eine Zahl k.
Frage/Problem: Gibt es einen Weg von s nach t der Länge $\leq k$?
Hinweis: Das Kürzeste-Wege-Problem ist nicht NP-vollständig.

46113

Theoretische Informatik (nicht vertieft)

Herbst 2012

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2012**	**46113**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **9**

Bitte wenden!

Thema Nr. 1

Aufgabe (Komplexitätstheorie):

Sei *ImpF* die Menge aller aussagenlogischen Formeln, die ausschließlich mit den Konstanten 0 (*false*) und 1 (*true*), logischen Variablen xi mit $i \in \mathbb{N}$ und der Implikation $\Rightarrow$ als Operationszeichen aufgebaut sind, wobei auch Klammern zugelassen sind. Insbesondere sind in *ImpF* nicht die logischen Operatoren $\neg$ (not), $\wedge$ (and) und $\vee$ (or) erlaubt.

Wir betrachten das Problem $ImpSAT$:

Gegeben: $F \in ImpF$.

Problem: Ist F erfüllbar, d. h., gibt es eine Belegung der Variablen mit Konstanten 0 oder 1, so dass F den Wert 1 annimmt?

Zeigen Sie: $ImpSAT$ ist NP-vollständig.

Sie dürfen benützen, dass das SAT-Problem (Erfüllbarkeitsproblem der Aussagenlogik)

NP-vollständig ist.

Fortsetzung nächste Seite!

Aufgabe (Berechenbarkeitstheorie):

1. Zeigen Sie, dass die Abstandsfunktion $dist\ (m,n) = |m-n|$ primitiv rekursiv ist. Dabei bezeichnet "$-$" die ganzzahlige Subtraktion!

2. Zeigen Sie, dass die Funktion $qsum(n) = \sum_{i=0}^{n} i^2$ primitiv rekursiv ist!

Hinweis: Sie dürfen zusätzlich zu den Basisfunktionen der primitiven Rekursion die folgenden Funktionen als primitiv rekursiv annehmen: Addition $(m+n)$, Multiplikation $(m*n)$, modifizierte Subtraktion $(m \dot{-} n = \begin{cases} m-n, falls\ n<m \\ 0,\ sonst \end{cases})$ und Gleichheit $(m=n)$. Sie dürfen die erweiterte Komposition und das erweiterte rekursive Definitionsschema verwenden!

Fortsetzung nächste Seite!

Aufgabe (Berechenbarkeitstheorie):

Sei $\Sigma = \{0,1\}$ ein Alphabet und $f : \Sigma^* \to \Sigma^*$ die Funktion, die jede 0 aus einem Wort löscht (z.B. gilt $f(00100110) = 111$). Formal ist f durch die folgenden Gleichungen definiert:

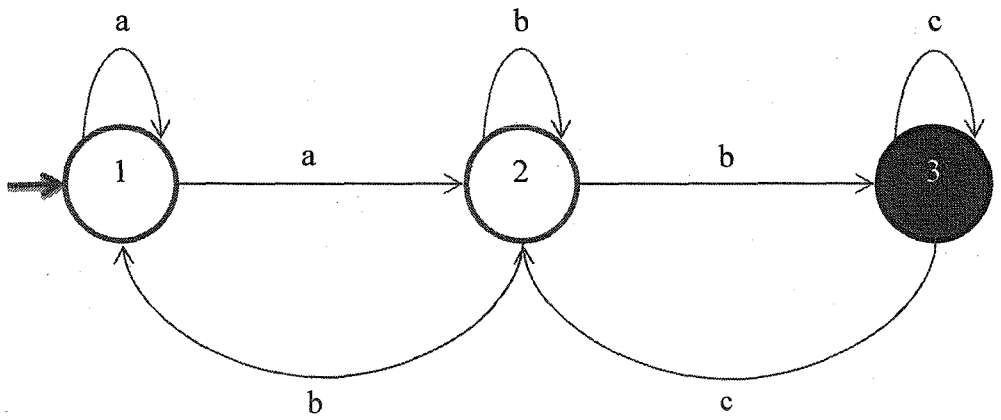
$$f(\varepsilon) = \varepsilon, \quad f(0w) = f(w), \quad f(1w) = 1f(w)$$

Konstruieren Sie eine 1-Band-Turingmaschine M, die f berechnet! Geben Sie M als Tupel an und beschreiben Sie die Zustandsübergangsfunktion als Graph, Tabelle oder Liste von Gleichungen! Kommentieren Sie Ihre Konstruktion durch eine informelle Beschreibung Ihrer Lösungsidee!

Fortsetzung nächste Seite!

Aufgabe (Formale Sprachen und Automatentheorie):

Seien $\Sigma = \{a,b,c\}$ und $A = (\{1,2,3\},\Sigma,\delta,1\{3\})$ ein nicht-deterministischer endlicher Automat, der durch das folgende Übergangsdiagramm definiert ist:



1. Geben Sie einen regulären Ausdruck r_A an, der $L(A)$ darstellt!
2. Berechnen Sie mithilfe der Potenzmengenkonstruktion einen deterministischen endlichen Automaten B mit $L(B) = L(A)$.
3. Sei $\Sigma_c = \Sigma \cup \{d\}$. Geben Sie einen endlichen Automaten C an mit $L(C) = \Sigma_c^* \setminus L(A)$.

Fortsetzung nächste Seite!

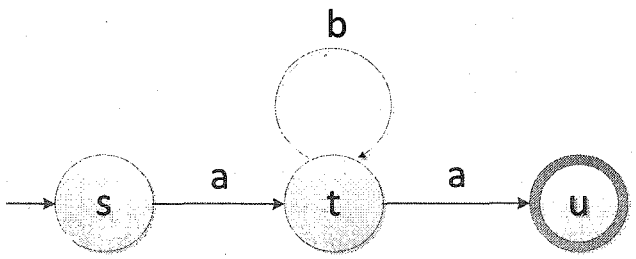
Aufgabe (Reguläre Ausdrücke und Endliche Automaten):

Für ein Alphabet Σ und eine Sprache $L \subseteq \Sigma^*$ definieren wir die Sprache

$$E(L) = \{x_1 x_3 \ldots x_{2n-1} \mid x_1, x_2, \ldots, x_{2n} \in \Sigma \wedge x_1 x_2 \ldots x_{2n} \in L\}$$

Damit ist $E(L)$ die Sprache aller Wörter, die dadurch entstehen, dass man bei einem Wort aus L geradzahliger Länge jedes zweite Zeichen entfernt.

1. Geben Sie reguläre Ausdrücke für $E(L((ab)^*))$ und $E(L(a|b|(aba)^*))$ an.
2. Gegeben sei folgender DFA N:



Geben Sie einen endlichen Automaten (DFA oder NFA) an, der $E(L(N))$ akzeptiert.

3. Zeigen Sie, z.B. mittels einer Automatenkonstruktion, die Gültigkeit folgender Aussage: Wenn L regulär ist, dann ist auch $E(L)$ regulär.

Thema Nr. 2

Aufgabe 1: Formale Sprachen

1. Sind die folgenden Sprachen regulär? Begründen Sie Ihre Antwort!
   (a) Die Menge aller Zeichenketten der Form $w \circ w^{rev}$, wobei w eine Zeichenkette aus Nullen und Einsen der Länge $|w| \leq 10$ ist und w^{rev} das Wort w *rückwärts* gelesen ist.
   (b) Die Menge aller Zeichenketten der Form $01^i01^j01^{i \cdot j}0$ für $i, j \geq 0$ über dem Alphabet $\Sigma = \{0, 1\}$.
2. Beweisen oder widerlegen Sie folgende Aussage:

   Sei X eine reguläre Sprache, dann ist auch jede Sprache $Y \subseteq X$ regulär.
3. Löschen und Einfügen in reguläre Sprachen
   (a) Sei L eine reguläre Sprache über dem Alphabet $\Sigma = \{a, b, c\}$. Zeigen Sie, dass die Sprache L', die entsteht, wenn man aus den Worten $w \in L$ alle Vorkommen des Zeichens c entfernt, ebenfalls regulär ist.
   (b) Sei L eine reguläre Sprache über dem Alphabet Σ und sei $x \notin \Sigma$. Zeigen Sie, dass die Sprache $L' \subseteq (\Sigma \cup \{x\})^*$ gegeben als

   $$L' = \{x^{i_1}w_1x^{i_2}w_2 \ldots x^{i_k}w_kx^{i_{k+1}} \mid w_1w_2 \ldots w_k \in L; w_l \in \Sigma; i_j \geq 0\}$$

   ebenfalls regulär ist.
4. Chomsky-Normalform
   (a) Für welche Grammatiken gibt es eine Chomsky-Normalform? Wann ist eine Grammatik in Chomsky-Normalform?
   (b) Geben Sie für folgende Sprache eine Grammatik an und bringen Sie sie in Chomsky-Normalform.

   $$L = \{a^nb^nc^m \mid n, m > 0\}$$

Fortsetzung nächste Seite!

Aufgabe 2: NP-Vollständigkeit

(a) Wann ist ein Problem NP-vollständig?

Folgendes Problem wird CLIQUE genannt und ist NP-vollständig:
gegeben: Ein Graph $G = (V, E)$, ein $k > 1$.
Frage: Hat G eine Clique (einen vollständigen Teilgraphen bei dem zwischen je zwei Knoten eine Kante besteht) mit $\geq k$ Knoten?

Folgendes Problem wird KNOTENÜBERDECKUNG genannt:
gegeben: Ein Graph $G = (V, E)$, ein $k > 1$.
Frage: Hat G eine überdeckende Knotenmenge $V' \subset V$ (für alle Kanten $\{u, v\} \in E$ gilt $u \in V' \vee v \in V'$) mit höchstens k Knoten?

(b) Zeigen Sie, dass KNOTENÜBERDECKUNG $\in$ NP.

(c) Zeigen Sie, dass
$G = (V, E)$ hat eine Clique mit $\geq k$ Knoten genau dann, wenn $\overline{G} = (V, \{\{u, v\} | u, v \in V, \{u, v\} \notin E\})$ eine überdeckende Knotenmenge von höchstens $|V| - k$ Knoten hat.

(d) Folgern Sie daraus, dass KNOTENÜBERDECKUNG NP-vollständig ist.

Fortsetzung nächste Seite!

Aufgabe 3: Entscheidbarkeit und Berechenbarkeit

(a) Was ist der Zusammenhang des Konzepts *Entscheidbarkeit* für Sprachen mit dem Konzept der *Berechenbarkeit* für Funktionen?

(b) Wann sagt man, dass ein Problem P *semi-entscheidbar* ist?

(c) Beweisen oder widerlegen Sie folgende Aussage:
Für eine konkrete Instanz I und ein beliebiges Entscheidungsproblem P (z.B. das Postsche Korrespondenzproblem) ist die Funktion f_I mit

$$f_I = \begin{cases} wahr & \text{falls } I \text{ eine } ja \text{ Instanz von } P \text{ ist,} \\ falsch & \text{sonst.} \end{cases}$$

berechenbar.

(d) Seien $A \subseteq \Sigma^*$ und $B \subseteq \Gamma^*$ Sprachen. Welche Eigenschaften muss eine Funktion $f : \Sigma^* \to \Gamma^*$ besitzen, um A auf B zu reduzieren?

(e) Betrachten Sie die Reduktion $A \leq B$,

i. Wenn A (semi-)entscheidbar ist, was ist über B bekannt?

ii. Wenn B (semi-)entscheidbar ist, was ist über A bekannt?

iii. Wenn A nicht entscheidbar ist, was ist über B bekannt?

iv. Wenn B nicht entscheidbar ist, was ist über A bekannt?

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2013

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Frühjahr 2013**	**46113**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **6**

Bitte wenden!

Thema Nr. 1

Aufgabe 1: Reguläre Sprachen I

1. Sei $\Sigma = \{a, b\}$. Für ein $w \in \Sigma^*$ und $x \in \Sigma$ bezeichnen wir mit $|w|_x$ die Anzahl der x in w. Sei $L = \{w \in \Sigma^* \mid |w|_a \text{ gerade}\}$. Geben Sie einen regulären Ausdruck α mit $L(\alpha) = L$ an.

2. Eine Grammatik heißt *rechts-linear*, wenn es für jede Produktion Nichtterminale A, B und ein Terminalsymbol x gibt, so dass die Produktion die Form $A \to xB$ oder $A \to \epsilon$ hat. Geben Sie eine rechtslineare Grammatik für die Sprache L aus Teilaufgabe 1 an.

3. Geben Sie einen endlichen Automaten (gegebenenfalls mit ε-Transitionen) an, der die Sprache $L(c(a|b^*c)^*c)$ erkennt.

Aufgabe 2: Reguläre Sprachen II

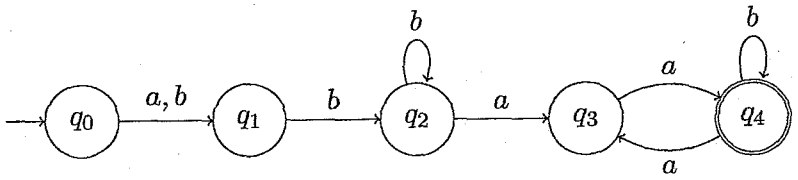
Sei Σ ein Alphabet. Für eine Sprache $L \subseteq \Sigma^*$ nennen wir

$$\begin{aligned} zieh(L) = \{a_1^{\ell_1} a_2^{\ell_2} \cdots a_n^{\ell_n} \mid\ & a_1^{k_1} a_2^{k_2} \cdots a_n^{k_n} \in L \wedge a_1, a_2, \ldots, a_n \in \Sigma \\ & \wedge 1 \leq \ell_1 \leq k_1 \wedge 1 \leq \ell_2 \leq k_2 \wedge \cdots \wedge 1 \leq \ell_n \leq k_n\} \end{aligned}$$

die Zusammenziehsprache von L.

Beispiel: Ist $L = \{aabbca, cc\}$, so ist $zieh(L) = \{abca, aabca, abbca, aabbca, c, cc\}$.

1. Sei L die Sprache, die durch den folgenden endlichen Automaten akzeptiert wird. Geben Sie einen nichtdeterministischen endlichen Automaten für $zieh(L)$ an.



2. Zeigen Sie: Ist L regulär, so ist $zieh(L)$ ebenfalls eine reguläre Sprache. Hinweis: Starten Sie mit einem endlichen Automaten für L und fügen Sie zusätzliche Übergänge ein. Erläutern Sie Ihre Konstruktion.

Fortsetzung nächste Seite!

Aufgabe 3: Kontextfreie Sprachen

1. Geben Sie eine kontextfreie Grammatik für die folgende Sprache an:

$$L = \{(abb)^n b^m (ab)^n \mid n \geq 0, m \text{ gerade}\}$$

2. Bringen Sie die Grammatik $G_1 = (N_1, \Sigma_1, P_1, S)$ in Chomsky-Normalform. Dabei sei $N_1 = \{S, T, U\}$, $\Sigma_1 = \{a, b, c\}$ und P_1 bestehe aus den folgenden Produktionen:

$$S \to aT \qquad T \to aTU \mid \varepsilon \qquad U \to bc$$

3. Betrachten Sie die Grammatik $G_2 = (N_2, \Sigma_2, P_2, S)$ mit $N_2 = \{S, T, U, A, B, C\}$ und $\Sigma_2 = \{a, b, c\}$, wobei P_2 aus den folgenden Produktionen besteht:

$$\begin{array}{lll} S \to AU \mid SS & T \to AB \mid BT \mid TC & U \to BA \mid BC \mid UB \\ A \to a & B \to b & C \to c \end{array}$$

   Verwenden Sie den CYK-Algorithmus um zu zeigen, dass das Wort *babc* nicht in $L(G_2)$ liegt.

4. Für ein $X \in N_2$ sei $G_X = (N_2, \Sigma_2, P_2, X)$ mit N_2, Σ_2 und P_2 aus der vorherigen Teilaufgabe. Geben Sie ein $X \in N_2$ an, so dass $babc \in L(G_X)$ gilt.

Aufgabe 4: Komplexitätstheorie

Mit EX bezeichnen wir die Menge aller aussagenlogischen Formeln, die ausschließlich aus den Konstanten 0 und $\uparrow$ und logischen Variablen aufgebaut sind (Klammern sind erlaubt). Dabei ist $x \uparrow y$ so definiert, dass es die gleiche Wahrheitstafel wie $x \to \neg y$ hat.

Wir betrachten das Problem EXSAT:

Gegeben: $F \in EX$

Problem: Ist F erfüllbar, d.h., gibt es eine Belegung der Variablen mit 0 oder 1, so dass F den Wert 1 annimmt?

1. Begründen Sie: EXSAT liegt in NP.

2. Finden Sie eine Formel $F(x)$ in EX, die äquivalent zu $\neg x$ ist und x nur einmal enthält.

3. Zeigen Sie: EXSAT ist NP-hart. Sie dürfen dazu benutzen, dass das SAT-Problem NP-hart ist.

Fortsetzung nächste Seite!

Aufgabe 5: Entscheidbarkeit

Sei $\Sigma = \{0, 1\}$. Wir nehmen an, dass jedes $w \in \Sigma$ eine Turingmaschine kodiert und bezeichnen diese TM mit M_w. Die von M_w berechnete Funktion bezeichnen wir mit φ_w.

1. Welche der folgenden Sprachen sind entscheidbar? Begründen Sie kurz.
   a) $L_1 := \{w \in \Sigma^* \mid \varphi_w(x) = \varepsilon$ für alle Eingaben $x\}$.
   b) $L_2 := \{w \in \Sigma^* \mid \exists w'$, so dass $\varphi_w(x) = \varphi_{w'}(x)$ für mindestens ein $x\}$.
   c) $L_3 := \{w \in \Sigma^* \mid M_w$ hat mindestens 12 Zustände$\}$
2. Semientscheidbarkeit
   a) Eine Sprache ist *semientscheidbar* genau dann, wenn sie *rekursiv aufzählbar* ist. Definieren Sie einen dieser beiden Begriffe.
   b) Geben Sie eine semientscheidbare Sprache an, die nicht entscheidbar ist. Begründen Sie, dass die Sprache diese beiden Eigenschaften besitzt.
3. Konstruieren Sie eine Turingmaschine M mit Eingabealphabet Σ, die alle 0-en aus dem Eingabewort entfernt. Das heißt, M berechnet die folgende Funktion $\Sigma^* \to \Sigma^*$:

$$f(\varepsilon) = \varepsilon \qquad f(1w) = 1f(w) \qquad f(0w) = f(w)$$

Beispiel: Für die Eingabe 0110001 soll M die Ausgabe 111 liefern.

Thema Nr. 2

Aufgabe 1: regulär

Sei L die Menge aller Worte über dem Alphabet {a,b,c},
bei denen der zweite und der zweitletzte Buchstabe übereinstimmen.

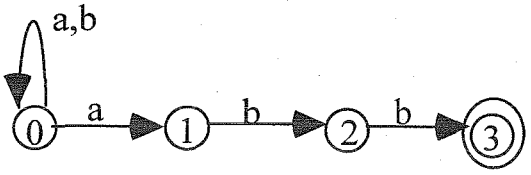
a) Listen Sie alle Worte aus L der Länge 3 auf.

b) Beschreiben Sie L durch einen regulären Ausdruck.

c) Geben Sie für L einen deterministischen endlichen Automaten (DFA) an.

Aufgabe 2 : endliche Automaten

a) Konstruieren Sie zum nachfolgenden nichtdeterministischen endlichen Automaten (NFA) A einen äquivalenten DFA.
A hat die Zustände {0,1,2,3}, den Anfangszustand 0, den Endzustand 3 und die angegebenen Zustandsübergänge



b) Minimieren Sie Ihren DFA oder, falls er bereits minimal ist, begründen Sie warum.

Aufgabe 3 : regulär

Sei $L = \{ a^n b^{2n} \mid n > 0\}$

a) Zeigen Sie, dass L kontextfrei ist.
b) Zeigen Sie, dass L nicht regulär ist.

Fortsetzung nächste Seite!

Aufgabe 4: kontextfrei

Sei $G = (N, T, S, P)$ eine kontextfreie Grammatik mit den Produktionen

$S \rightarrow aaSb$, $S \rightarrow A$, $A \rightarrow aAb$, $A \rightarrow ab$.

a) Konstruieren Sie eine äquivalente kontextfreie Grammatik G' in Chomsky-Normalform.

b) Beschreiben Sie L in Worten in der Form:
L besteht aus der Menge aller Worte w über dem Alphabet {a,b} mit w = ...

Aufgabe 5: Turingmaschinen

Konstruktion von Turingmaschinen

a) Konstruieren Sie eine (k-Band) Turingmaschine M für die Sprache
$PAL = \{a^n \; c \; a^n \; c \; a^n \mid n > 0\}$

b) Welche Zeit-Komplexität hat Ihre Turingmaschine M.

Aufgabe 6:

Was ist mit dem P – NP – Problem gemeint?
Definieren Sie P und NP und beschreiben Sie, was mit P versus NP gemeint ist.

46113

Theoretische Informatik (nicht vertieft)

Herbst 2013

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____	**Herbst 2013**	**46113**
Kennwort: _____		
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **8**

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:

Reguläre Sprachen I

Sei $\Sigma = \{a, b\}$ und L_1 die Sprache bestehend aus allen Wörtern gerader Länge über dem Alphabet Σ. Sei L_2 die Sprache bestehend aus allen Wörtern über Σ, in denen b genau zweimal vorkommt.

a) Geben Sie jeweils einen regulären Ausdruck für L_1 und L_2 an.

b) Eine Grammatik heißt *rechtslinear*, wenn es für jede Produktion Nichtterminale A, B und ein Terminalsymbol x gibt, so dass die Produktion die Form $A \to xB$ oder $A \to \epsilon$ hat. Geben Sie eine rechtslineare Grammatik für die Sprache $L_1 \cap L_2$ an.

c) Geben Sie einen endlichen Automaten (gegebenenfalls mit ε-Transitionen) an, der die Sprache $L(a(b^*|c)^*(a|b))$ erkennt.

Fortsetzung nächste Seite!

Aufgabe 2:

Reguläre Sprachen II

Sei $\Sigma = \{0, 1\}$. Wir definieren die Inversion $\overline{u}$ von Wörtern $u \in \Sigma^*$ wie folgt:

$$\begin{aligned} \overline{\varepsilon} &:= \varepsilon \\ \overline{0v} &:= 1\overline{v} \text{ für } v \in \Sigma^* \\ \overline{1v} &:= 0\overline{v} \text{ für } v \in \Sigma^* \end{aligned}$$

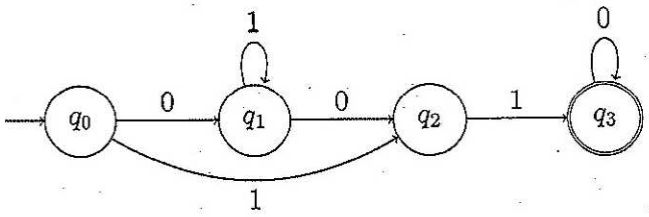
Für eine Sprache $L \subseteq \Sigma^*$ definieren wir

$$\mathsf{inverse_from}(L) := \{a_1 \cdots a_n \mid \exists k. \in \mathbb{N} a_1 \cdots a_k \overline{a_{k+1} \cdots a_n} \in L \wedge 0 \leq k \leq n\}$$

die Sprache, die durch Invertieren der Wörter aus L ab beliebigen Positionen hervorgeht.

Beispiel: Ist $L = \{000, 11\}$, so ist $\mathsf{inverse_from}(L) = \{000, 001, 011, 111, 11, 10, 00\}$.

a) Sei L die Sprache, die durch den folgenden endlichen Automaten akzeptiert wird. Geben Sie einen endlichen Automaten (gegebenenfalls mit ε-Transitionen) für $\mathsf{inverse_from}(L)$ an.
Hinweis: Starten Sie mit einem endlichen Automaten für L und fügen Sie zusätzliche Zustände und Übergänge ein.



b) Zeigen Sie: Ist $L \subseteq \Sigma^*$ regulär, so ist $\mathsf{inverse_from}(L)$ ebenfalls eine reguläre Sprache.
Hinweis: Verallgemeinern Sie Ihre Konstruktion von Teilaufgabe 1.
Erläutern Sie Ihre Konstruktionsidee!

Fortsetzung nächste Seite!

Aufgabe 3:

Kontextfreie Sprachen

a) Geben Sie eine kontextfreie Grammatik für die folgende Sprache an:

$$L = \{a^n b^m c^\ell b^{2n} \mid 0 \leq n, 0 \leq m < \ell\}$$

b) Sei $G_1 = (N_1, \Sigma_1, P_1, S)$ mit $N_1 = \{S, T\}$, $\Sigma_1 = \{a, b, c, d\}$ und P_1 bestehe aus den folgenden Produktionen:

$$S \to TaS \mid T \qquad T \to bSc \mid d$$

i) Zeigen Sie $dabdc \in L(G_1)$.

ii) Bringen Sie die Grammatik G_1 in Chomsky-Normalform.

Fortsetzung nächste Seite!

Aufgabe 4:

Berechenbarkeit und Entscheidbarkeit

Sei $\Sigma = \{0, 1\}$. Wir nehmen an, dass jedes $w \in \Sigma^*$ eine Turingmaschine kodiert und bezeichnen diese TM mit M_w. Die von M_w berechnete Funktion bezeichnen wir mit φ_w.

a) Welche der folgenden Mengen sind entscheidbar? Begründen Sie kurz.

i) $L_1 := \{w \in \Sigma^* \mid \text{es gibt eine Eingabe, für die } M_w \text{ das leere Wort ausgibt}\}$

ii) $L_2 := \{w \in \Sigma^* \mid \exists n \in \mathbb{N} \text{ mit } w = 0^n 1^n\}$

iii) $L_3 := \{w \in \Sigma^* \mid \forall x \in \Sigma^*. x \in L(0^*1^*) \leftrightarrow \varphi_w(x) = 1\}$

b) Ist die folgende Funktion berechenbar? Begründen Sie kurz.

$$h(w) = \begin{cases} 1 & \text{wenn } \varphi_w(01) = 10 \\ 0 & \text{sonst} \end{cases}$$

c) Konstruieren Sie eine Turingmaschine M mit Eingabealphabet Σ, die jedes Vorkommen von 1 im Eingabewort verdoppelt. Das heißt, M soll die folgende Funktion $f : \Sigma^* \to \Sigma^*$ berechnen:

$$f(\varepsilon) = \varepsilon \qquad f(1w) = 11f(w) \qquad f(0w) = 0f(w)$$

Beschreiben Sie auch Ihre Konstruktionsidee!
Beispiel: Für die Eingabe 01101 soll M die Ausgabe 01111011 liefern.

Fortsetzung nächste Seite!

Aufgabe 5:

Komplexitätstheorie

Wir betrachten das Problem, in einer Schulklasse Gruppenarbeit durchzuführen. Es gibt mehrere *Arbeitsthemen*; jedem *Schüler* wird ein Thema zugeordnet und alle Schüler mit dem gleichen Thema arbeiten in einer Gruppe. Allerdings weiß man, dass wenn gewisse *Teilgruppen* von Schülern vollständig einem Arbeitsthema zugeordnet sind, sie sich gegenseitig und somit die ganze Gruppe stören. Man möchte durch eine geeignete Zuordnung von Schülern zu Arbeitsthemen sicherstellen, dass keine Störungen auftreten.

Formal betrachtet definieren wir das Problem GRUPPENARBEIT wie folgt:

Gegeben: $(S, \mathcal{T}, A)$:
S ... Menge von Schülern
$\mathcal{T}$ ... störende Teilgruppen, $\mathcal{T} \subseteq \mathcal{P}(S)$
A ... Menge der Arbeitsthemen

Problem: Existiert eine Zuordnung $g : S \to A$, sodass aus jeder störenden Teilgruppe mindestens zwei Schüler unterschiedlichen Arbeitsthemen zugeordnet sind:

$$\forall T \in \mathcal{T} : \ \exists t_1 \in T : \ \exists t_2 \in T : \ g(t_1) \neq g(t_2)$$

a) Begründen Sie: GRUPPENARBEIT liegt in NP.

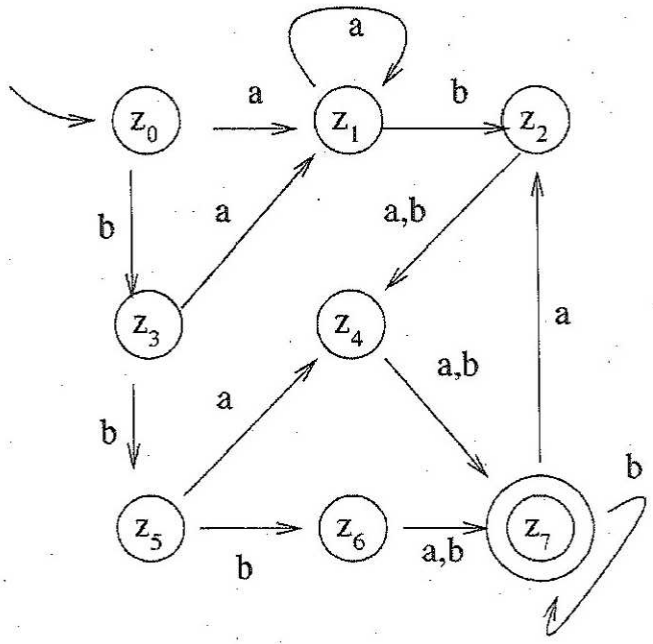
b) Geben Sie eine polynomielle Reduktion von 3-COL auf GRUPPENARBEIT an. Begründen Sie kurz die polynomielle Laufzeit und die Korrektheit und Vollständigkeit der Reduktion.
Hinweis: 3-COL ist definiert als das Problem, für einen gegebenen Graphen (V, E) zu entscheiden, ob die Knoten V so mit drei Farben eingefärbt werden können, dass keine durch Kanten in E benachbarte Knoten derselben Farbe zugeordnet sind.

c) Zeigen Sie: GRUPPENARBEIT ist NP-vollständig.
Hinweis: Sie dürfen verwenden, dass 3-COL NP-hart ist.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Teilaufgabe I: Minimierung und Äquivalenzen Gegeben ist folgender deterministische endliche Automat (DEA) A über dem Alphabet $\Sigma = \{a, b\}$.



a) Konstruieren Sie den zugehörigen Minimalautomaten und dokumentieren Sie Ihre Schritte auf geeignete Weise.

b) Geben Sie $L(A)$ als regulären Ausdruck an.

c) Geben Sie reguläre Ausdrücke für die Äquivalenzklassen der Myhill-Nerode Äquivalenzrelation $\sim_{L(A)}$ an. Erinnerung: Diese Klassen entsprechen den Zuständen des Minimalautomaten.

d) Sei $B = (Z, \Sigma, \delta, z_0, E)$ der von Ihnen konstruierte Minimalautomat. Wir definieren die Beobachtungsgleichheit $\simeq_B$ auf Σ^* durch $u \simeq_B v \Longleftrightarrow \forall z \in Z{:}\delta(z, u) = \delta(z, v)$.

Sollten Sie die vorhergehenden Aufgaben nicht gelöst haben, so nehmen Sie für B einen DEA Ihrer Wahl mit $L(B) = (a \mid b)b^*a(a \mid b)b^*$. Bitte vermerken Sie dann, dass Sie diese Option gewählt haben und geben Sie Ihren DEA B explizit an.

- Begründen Sie, dass $\simeq_B$ eine Äquivalenzrelation ist.
- Geben Sie zwei verschiedene Wörter an, die in derselben $\simeq_B$-Klasse liegen.
- Geben Sie zwei Wörter an, die in unterschiedlichen Klassen liegen.
- Begründen Sie, dass $\simeq_B$ nur endlich viele Klassen hat.

Fortsetzung nächste Seite!

Teilaufgabe II: Berechenbarkeit Bekanntlich ist das Halteproblem H_0 definiert als die Menge aller Codes von Turingmaschinen, die bei leerer Eingabe halten. I.Z.: $H_0 = \{e \mid M_e(\epsilon)\ \textit{hält}\}$.

a) Begründen Sie, dass H_0 rekursiv aufzählbar (= semi-entscheidbar) ist, das Komplement $\overline{H_0} = \mathbb{N} \setminus H_0$ aber nicht.

b) Eine Turingmaschine M kann man effektiv zu einer anderen Maschine $tr(M)$ umbauen, sodass gilt: $tr(M)$ hält bei Eingabe $t \in \mathbb{N}$, genau dann, wenn M bei leerer Eingabe mindestens t Schritte rechnet. Begründen Sie dies.

c) Es sei $Tot = \{e \mid M_e$ hält für alle Eingaben $x \in \mathbb{N}\}$ das *Totalitätsproblem*.
Welche der folgenden Aussagen sind richtig? Begründen Sie jeweils Ihre Antwort!

- Die Konstruktion tr liefert eine Reduktion $\overline{H}_0 \leq Tot$.
- Die Konstruktion tr liefert eine Reduktion $Tot \leq \overline{H}_0$.
- Die Konstruktion tr liefert eine Reduktion $Tot \leq H_0$.
- Das Problem Tot ist semi-entscheidbar.
- Das Problem Tot ist nicht rekursiv aufzählbar.

Teilaufgabe III: Komplexität Beim Problem MAX2SAT ist eine Menge von Zweierklauseln und eine Zahl $k \in \mathbb{N}$ gegeben. Gefragt ist, ob eine Variablenbelegung existiert, die mindestens k von den Zweierklauseln erfüllt. Erinnerung: Zweierklausel: Disjunktion zweier Literale; Literal: negierte oder nichtnegierte aussagenlogische Variable.

Ist zum Beispiel $\mathcal{C} = \{X \vee \neg Y, Y \vee \neg Z, Z \vee \neg X, X \vee Y, \neg X \vee \neg Y\}$, so ist $\mathcal{C}, k = 4$ eine positive Instanz von MAX2SAT, $\mathcal{C}, k = 5$ aber eine negative Instanz.

Beim Problem CLIQUE hingegen ist ein ungerichteter Graph $G = (V, E)$ und $j \in \mathbb{N}$ gegeben und es ist gefragt, ob es in G mindestens j paarweise verbundene Knoten gibt (eine Clique der Größe j). Das Problem CLIQUE ist bekanntlich NP-vollständig.

Hat man einen ungerichteten Graphen $G = (V, E)$ gegeben, so konstruiert man eine Menge von Zweierklauseln wie folgt: Variablen sind die Knoten von G. Für je zwei Knoten v, v' mit $(v, v') \notin E$ gibt man eine Klausel $\neg v \vee \neg v'$ aus und außerdem für jedes $v \in V$ die Klausel $v \vee v$. Insgesamt also $|V| + d$ Klauseln, wobei d die Zahl der Nicht-Kanten in G ist.

a) Begründen Sie folgendes: Hat G eine Clique der Größe j, so kann man mindestens $d + j$ viele dieser Klauseln simultan erfüllen.

Es gilt auch die Umkehrung dieser Aussage (können irgend $d + j$ der Klauseln simultan erfüllt werden, so hat G eine Clique der Größe j), das dürfen Sie im folgenden ohne Begründung voraussetzen.

b) Was kann aus dieser Konstruktion nun gefolgert werden? Begründen Sie jeweils Ihre Antwort:

- Diese Konstruktion liefert eine Reduktion CLIQUE $\leq_p$ MAX2SAT.
- Diese Konstruktion liefert eine Reduktion MAX2SAT $\leq_p$ CLIQUE.
- Die Konstruktion liefert ein Entscheidungsverfahren für MAX2SAT mit polynomieller Laufzeit.
- MAX2SAT ist NP-vollständig.

46113

Theoretische Informatik (nicht vertieft)

Frühjahr 2014

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____		
Kennwort: _____	**Frühjahr 2014**	**46113**
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): **2**

Anzahl der Druckseiten dieser Vorlage: **7**

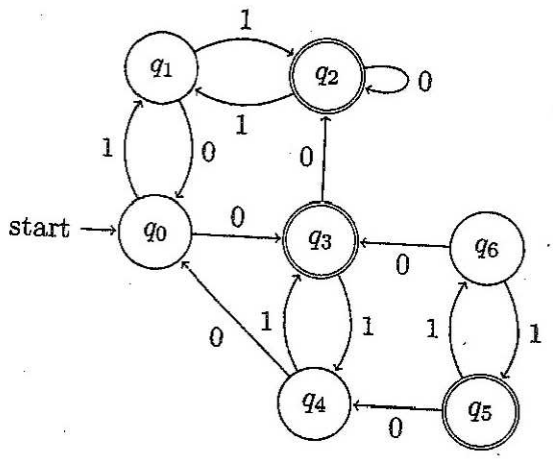
Bitte wenden!

Thema Nr. 1

Aufgabe 1: Reguläre Sprachen I

Sei $\Sigma = \{a, b\}$ und L_1 die Sprache aller Wörter über Σ, in denen das Wort ab nicht vorkommt. Sei weiterhin L_2 die Sprache aller Wörter über Σ, in denen b genau zwei mal vorkommt.

1. Geben Sie jeweils einen regulären Ausdruck für L_1 und L_2 an.
2. Eine Grammatik heißt *rechtslinear*, wenn es für jede Produktion Nichtterminale A, B und ein Terminalsymbol x gibt, so dass die Produktion die Form $A \to xB$ oder $A \to \varepsilon$ hat. Geben Sie eine rechtslineare Grammatik für die Sprache $L_2 \setminus L_1$ an.
3. Geben Sie einen minimalen deterministischen endlichen Automaten an, der die gleiche Sprache erkennt wie der unten aufgeführte Automat! Geben Sie zu jedem Zustand des minimalen Automaten an, zu welchen Zuständen des originalen Automaten er gehört!



Fortsetzung nächste Seite!

Aufgabe 2: Kontextfreie Sprachen

1. Betrachten Sie die Grammatik $G = (N, \Sigma, P, S)$ mit $N = \{S, T, U, V, A, C\}$ und $\Sigma = \{a, b, c\}$. Die Menge P besteht aus den folgenden Produktionen:

$$\begin{array}{ll} S \to AT & V \to a \mid b \mid CU \\ T \to UA & A \to a \\ U \to CV \mid AT \mid VV & C \to c \end{array}$$

   Zeigen Sie mit dem CYK-Algorithmus, dass das Wort $acca$ nicht in $L(G)$ liegt.

2. Zeigen Sie mit dem Pumping-Lemma für kontextfreie Sprachen, dass die folgende Sprache nicht kontextfrei ist:

   $\mathrm{L} = \{\ \mathrm{a}^n b a^n b a^n \mid n \in \mathbb{N}\}$

Fortsetzung nächste Seite!

Aufgabe 3: Komplexitätstheorie

Wir betrachten das Problem des Handlungsreisenden (TSP). Zur Erinnerung: Für eine Menge von Städten S und eine Entfernungsfunktion d zwischen diesen Städten stellen wir die folgende Frage: Existiert eine Route, die jede Stadt genau einmal besucht und dann zum Ausgangsort zurückkehrt, so dass die insgesamt zurückgelegte Entfernung höchstens x ist?

Formal ist das Problem TSP wie folgt definiert:

Gegeben: (S, d, x)
S ... Menge von Städten
d ... Totale Distanzfunktion $d : S \times S \to \mathbb{R}$
x ... Maximal erlaubte Gesamtdistanz

Problem: Existiert eine Folge $s_1, \ldots, s_n$ mit $|S| = n$, so dass alle s_i unterschiedlich sind und die Gesamtentfernung $\left(\sum_{1 \leq i < n} d(s_i, s_{i+1})\right) + d(s_n, s_1)$ höchstens x ist?

1. Begründen Sie: TSP liegt in NP.
2. Geben Sie eine polynomielle Reduktion von HAMILTON auf TSP an! Das Problem HAMILTON ist die Frage, ob ein Graph einen *Hamiltonpfad* besitzt. Ein Hamiltonpfad ist ein Weg im Graphen, der jeden Knoten genau einmal enthält.

   Formal ist das Problem HAMILTON wie folgt definiert:

   Gegeben: Graph $G = (E, K)$:
   E ... Menge von Ecken
   K ... Menge von Kanten

   Problem: Existiert ein Hamiltonpfad in G, d.h., existiert eine Folge $e_1, \ldots, e_n$ mit $|E| = n$, so dass die e_i alle unterschiedlich sind und für alle $1 \leq i < n$ gilt: $(e_i, e_{i+1}) \in K$?

   Hinweis: Wählen Sie die Distanzfunktion d so, dass Sie aus der Distanz ablesen können, wie viele Kanten des Pfades nicht in K liegen.
3. Zeigen Sie: TSP ist NP-vollständig.

   Hinweis: Sie dürfen verwenden, dass HAMILTON *NP-hart ist.*

Fortsetzung nächste Seite!

Aufgabe 4: Berechen- und Entscheidbarkeit

1. Sei $\Sigma = \{0, 1\}$. Wir nehmen an, dass jedes $w \in \Sigma^*$ eine Turingmaschine kodiert und bezeichnen diese TM mit M_w. Die von M_w berechnete Funktion bezeichnen wir mit φ_w.

   Welche der folgenden Mengen sind entscheidbar? Begründen Sie kurz!

   a) $L_1 = \{w \in \Sigma^* \mid M_w$ terminiert für alle Eingaben und $|w| < 10^4\}$

   b) $L_2 = \{w \in \Sigma^* \mid \varphi_w(000) = \varphi_\varepsilon(111)\}$

   c) $L_3 = \{w \in \Sigma^* \mid M_w$ terminiert nicht auf 01 oder $\varphi_w(01) \in L((0+1)^*)\}$

2. Konstruieren Sie eine Turingmaschine M mit Eingabealphabet $\Sigma = \{0, \#\}$, die für eine Eingabe $\#^m w$ mit $w \in \{0\}^* \setminus \{\varepsilon\}$ und $m \in \mathbb{N}$ die Ausgabe $w(\#w)^m$ liefert! Für alle anderen Eingaben darf sich M beliebig verhalten.

   Beschreiben Sie auch Ihre Konstruktionsidee!

   Beispiel: Für die Eingabe ##00 soll M die Ausgabe 00#00#00 liefern.

Thema Nr. 2

Annahmen:

Sie dürfen als bekannt und bewiesen voraussetzen:
Die Sprache $\{a^n b^n \mid n \geq 1\}$ ist nicht regulär.
Die Sprache $\{a^n b^n c^n \mid n \geq 1\}$ ist nicht kontextfrei.
Um zu zeigen, dass eine Sprache L regulär (kontextfrei) ist, reicht die Angabe einer entsprechenden Beschreibung (Automat, Grammatik, Ausdruck).
Sie müssen nicht mehr zeigen, dass Ihre Beschreibung korrekt ist und genau die vorgegebene Sprache beschreibt.

Aufgabe 1: reguläre Mengen

Sei L die Menge aller Worte über dem Alphabet {a,b}, bei denen das zweite und letzte Zeichen übereinstimmen.

a) Geben Sie alle Worte bis zur (einschließlich) Länge zwei an!
b) Beschreiben Sie L
   i) durch einen regulären Ausdruck
   ii) durch einen deterministischen endlichen Automaten A!

Aufgabe 2: regulär oder nicht

Sei L die Sprache $L = \{ a^n b^m a^n \mid n,m \geq 1\}$.
Ist L regulär oder nicht?
Begründen Sie Ihre Antwort durch die Angabe einer passenden Beschreibung für L oder führen Sie den Nachweis, dass L nicht regulär sein kann!

Aufgabe 3: Abschlusseigenschaften

Begründen Sie:
Wenn L und L' reguläre Sprachen sind, dann ist auch die Vereinigung $L \cup L'$ regulär.

Aufgabe 4: kontextfrei

Zeigen Sie, dass die Sprache L (über den Zeichen {a,b,c}) kontextfrei ist!
$L = \{(ab)^n \; c \; (ab)^n \mid n \geq 1\}$.

Fortsetzung nächste Seite!

Aufgabe 5: Chomsky Normalform und Test

a) Konstruieren Sie zur folgenden Grammatik G = ({S, A}, {a,b}, S, P) eine äquivalente Grammatik
G' in Chomsky Normalform.
Die Produktionen von G sind:
S –> a S b S, S –> X, X –> a X b, X –> ab.

b) Zeigen Sie (durch Angabe einer Ableitung),
dass das Wort w = a a b b a a b b in L(G) bzw. L(G') liegt.

Aufgabe 6: Berechenbarkeit

Gegeben sei die Sprache PAL = { w c w^{rev} | w ∈ {a,b}* } über dem Alphabet {a,b,c}.
w^{rev} ist w gespiegelt.

a) Konstruieren Sie eine Turingmaschine (auch Mehrband oder nichtdeterministisch), die die Sprache PAL erkennt!
Beschreiben Sie, wie Ihre Turingmaschine arbeitet!

b) Welche Zeitkomplexität in O-Notation hat Ihre Turingmaschine?

Aufgabe 7: Komplexität

Warum ist die Sprache SAT der erfüllbaren booleschen Ausdrücke in konjunktiver Normalform in NP?

Erläutern Sie, was man für einen Ausdruck testen muss!
Nicht gefragt ist, warum das Problem sogar NP-vollständig ist.

46113

Theoretische Informatik (nicht vertieft)

Herbst 2014

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____		
Kennwort: _____	Herbst 2014	46113
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 4

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:

Wir fixieren das Alphabet $\Sigma = \{0, 1\}$ und definieren $L \subseteq \Sigma^*$ durch

$$L = \{w \mid \text{in } w \text{ kommt das Teilwort 0010 vor}\}$$

a) Zeigen Sie, dass L regulär ist.

b) Vervollständigen Sie durch Hinzufügen von Kanten und Angabe der Endzustände folgendes Diagramm zu einem deterministischen Automaten für L. Bitte das Diagramm auf Ihrem Arbeitspapier noch einmal abschreiben.

$$\rightarrow q_0 \xrightarrow{0} q_1 \xrightarrow{0} q_2 \xrightarrow{1} q_3 \xrightarrow{0} q_4$$

c) Zeigen Sie durch Ausführung des Minimierungsalgorithmus, dass dieser Automat minimal ist.

d) Geben Sie die Äquivalenzklassen der Myhill-Nerode-Äquivalenz von L durch Repräsentanten an. (Diese Äquivalenz ist definiert durch $x \sim_L y \iff \forall u : xu \in L \Leftrightarrow yu \in L$.)

Geben Sie zu zwei Klassen Ihrer Wahl neben dem gewählten Repräsentanten noch ein weiteres Element an.

Geben Sie für die Klasse, in der sich das leere Wort befindet, einen regulären Ausdruck an. Hilfe: Diese Klasse ist identisch mit $\{w \mid 10001w \in L\}$.

Aufgabe 2:

Ordnen Sie die folgenden formalen Sprachen bestmöglich in die Chomsky-Hierarchie ein und geben Sie eine ausreichende Begründung an:

a) $L_1 = \{\, a^n b a^n \mid n \geq 1 \,\}$

b) $L_2 = \{\, a^n b^n c^n \mid n \geq 1 \,\}$

c) L_3 sei die Menge aller syntaktisch korrekten Java-Programme, die ohne Eingabe terminieren.

Aufgabe 3:

Bekanntlich sind beim SUBSET-SUM Problem natürliche Zahlen $a_1, \ldots, a_n$ und eine Zielzahl b gegeben und es ist gefragt, ob eine Teilmenge $I \subseteq \{1, \ldots, n\}$ existiert, sodass $\sum_{i \in I} a_i = b$. Dieses Problem ist bekanntermaßen NP-vollständig.

a) Zeigen Sie, dass die Variante GSUBSET, bei der die Zahlen a_i und b allesamt gerade sein müssen, ebenfalls NP-vollständig ist.

b) In welche Komplexitätsklasse fällt die Variante ZSUBSET, bei der die a_i paarweise verschieden und Zweierpotenzen sind? Hilfe: Betrachten Sie die Binärdarstellung von b.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:

Sei $\Sigma = \{a, b\}$, sei $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$.

a) Sei

$$L_1 = \{a^n b^m \mid n, m \in \mathbb{N}_0, n > m \geq 0\} \ .$$

Beispiele: $aab \in L_1$, $aaaabb \in L_1$, $a \in L_1$.

i) Zeigen Sie, dass L_1 kontextfrei ist, indem Sie eine kontextfreie Grammatik G angeben mit $L(G) = L_1$, und

ii) begründen Sie, warum Ihre Grammatik *genau* die Sprache L_1 erzeugt.

b) Formulieren Sie das Pumping-Lemma für reguläre Sprachen:

„Sei L eine reguläre Sprache über dem Alphabet Σ. Dann gibt es ...“

c) Zeigen Sie mit Hilfe des Pumping-Lemmas für reguläre Sprachen, dass die Sprache L_1 nicht regulär ist.

Aufgabe 2:

a) Definieren Sie die zum Halteproblem für Turing-Maschinen bei fester Eingabe $m \in \mathbb{N}_0 = \{0, 1, 2, \ldots\}$ gehörende Menge H_m.
Es geht hier also um die Turing-Maschinen, die mit fest vorgegebener Eingabe m gestartet werden.

b) Gegeben sei das folgende Problem E:
Entscheide, ob es für die deterministische Turing-Maschine M_n mit der Gödelnummer n mindestens eine Eingabe $w \in \mathbb{N}_0$ gibt, so dass w eine *Primzahl* ist und die Maschine M_n gestartet mit w hält.

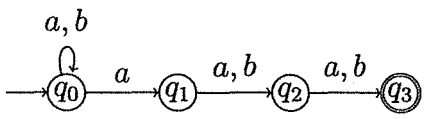
Zeigen Sie, dass E nicht entscheidbar ist. Benutzen Sie, dass H_m aus (a) für jedes $m \in \mathbb{N}_0$ nicht entscheidbar ist.

c) Zeigen Sie, dass das Problem E aus (b) partiell-entscheidbar (= rekursiv aufzählbar) ist.

Aufgabe 3:

a) Konstruieren Sie aus dem NEA $\mathcal{A}$ mit der Potenzmengenkonstruktion einen (deterministischen) EA, der dieselbe Sprache akzeptiert.

$\mathcal{A}$:



b) Welche Sprache wird von dem regulären Ausdruck $(a + b)^* a(a + b)$ beschrieben?

Fortsetzung nächste Seite!

Aufgabe 4:

a) Gegeben sei die folgende kontextfreie Grammatik $G = (V, \Sigma, P, S)$ mit $V = \{S, A, B, C, D\}$, $\Sigma = \{a, b, c\}$ und den Produktionen:

$$S \to A \mid aB \mid aC \qquad B \to S \mid Ba \qquad D \to b \mid bDD$$

$$A \to B \mid C \mid cAd \qquad C \to D \mid c$$

Konstruieren Sie eine äquivalente kontextfreie Grammatik G' ohne Kettenregeln, so dass $L(G') = L(G)$ gilt.

Hinweis: Eine Produktion heißt Kettenregel, wenn sie von der Form $A \to B$ für Variablen A und B ist.

b) Sei $G = (V, \Sigma, P, S)$ folgende kontextfreie Grammatik in Chomsky-Normalform:

$$S \to AB \mid BC \qquad B \to CC \mid b$$

$$A \to BA \mid a \qquad C \to AB \mid a$$

Sei $w = ababa$. Folgende Tabelle entsteht durch Anwendung des CYK-Algorithmus. Geben Sie die drei fehlenden Einträge $V(1,4)$, $V(2,5)$ und $V(1,5)$ an. Wie entnehmen Sie dieser Tabelle, dass $w \in L(G)$ ist?

a	b	a	b	a
{A,C}	{B}	{A,C}	{B}	{A,C}
{S,C}	{S,A}	{S,C}	{S,A}	
{B}	{S,C}	{B}		
V(1,4)	V(2,5)			
V(1,5)				

46113

Theoretische Informatik (nicht vertieft)

Herbst 2016

Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
Kennzahl: _____		
Kennwort: _____	Herbst 2016	46113
Arbeitsplatz-Nr.: _____		

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen

— Prüfungsaufgaben —

Fach: **Informatik (Unterrichtsfach)**

Einzelprüfung: **Theoretische Informatik**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 6

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

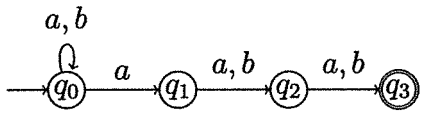
Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:

Reguläre Sprachen

a) Konstruieren Sie aus dem NEA $\mathcal{A}$ mit der Potenzmengenkonstruktion einen (deterministischen) EA, der dieselbe Sprache akzeptiert.

$\mathcal{A}$:



b) Beschreiben Sie möglichst einfach, welche Sprachen von den folgenden regulären Ausdrücken beschrieben werden:

1. $(a \mid b)^* a$
2. $(a \mid b)^* a (a \mid b)^* a (a \mid b)^*$
3. $(a \mid b)^* a (bb)^* a (a \mid b)^*$

Aufgabe 2:

Kontextfreie Grammatiken

Betrachten Sie die folgende kontextfreie Grammatik $G = (\{S, A, B, C\}, \{a, b, c\}, S, P)$ mit den Produktionen:

$$S \to AB \mid C$$

$$A \to a \mid \varepsilon$$

$$B \to BC$$

$$C \to c$$

a) Beschreiben Sie in Worten möglichst genau die von G erzeugte Sprache $L(G)$.

b) Geben Sie ohne Begründung an, welche Variablen in der Grammatik

- nützlich,
- erzeugend oder
- erreichbar

sind!

c) Ist die Grammatik in Chomsky-Normalform? Begründen Sie Ihre Antwort.

Fortsetzung nächste Seite!

Aufgabe 3:

Klassifizierung von Sprachen

Welche der folgenden Sprachen über dem Alphabet $\{a, b\}$ sind

a) regulär?

b) kontextfrei, aber nicht regulär?

c) nicht kontextfrei?

1. $L_1 = \{w \in \{a\}^* \mid \text{die Länge von } w \text{ ist } 2^n \text{ für eine natürliche Zahl } n\}$
2. $L_2 = \{a^k b^\ell \mid k \leq \ell\} \cup \{a^k b^\ell \mid k \geq \ell\}$
3. $L_3 = \{a^k b^\ell \mid k < \ell\} \cup \{a^k b^\ell \mid k > \ell\}$

Geben Sie zu Ihrer Antwort jeweils einen Beweis an. Ein Beweis durch Angabe eines Automaten oder eines regulären Ausdruckes ist erlaubt. Wird der Beweis durch die Angabe einer Grammatik geführt, so ist die Bedeutung der Variablen zu erläutern.

Aufgabe 4:

Probleme und Komplexität

Betrachten Sie die beiden folgenden Probleme:

VERTEXCOVER
Gegeben: Ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \{1, 2, 3, \ldots\}$.
Frage: Gibt es eine Menge $C \subseteq V$ mit $|C| \leq k$, so dass für jede Kante (u, v) aus E mindestens einer der Knoten u und v in C ist?

VERTEXCOVER3
Gegeben: Ein ungerichteter Graph $G = (V, E)$ und eine Zahl $k \in \{3, 4, 5, \ldots\}$.
Frage: Gibt es eine Menge $C \subseteq V$ mit $|C| \leq k$, so dass für jede Kante (u, v) aus E mindestens einer der Knoten u und v in C ist?

Geben Sie eine polynomielle Reduktion von VERTEXCOVER auf VERTEXCOVER3 an und begründen Sie anschließend, dass Ihre Reduktion korrekt ist.

Aufgabe 5:

Kurze Begründungen

Sind folgende Aussagen wahr oder falsch? Begründen Sie jeweils kurz.

a) Für jede reguläre Sprache R gibt es eine kontextfreie Grammatik G mit $L(G) = R$.

b) Falls $L = L_1 \cap L_2$ mit L_1 und L_2 kontextfrei, dann ist L eine unentscheidbare Sprache.

c) Falls r ein regulärer Ausdruck ist, so definiert r^* immer eine unendliche Sprache.

d) Eine entscheidbare Sprache ist niemals leer.

e) Alle Sprachen in NP sind entscheidbar.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:

Sei $\Sigma = \{0, 1\}$, sei $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$.

a) Sei

$$L_1 = \{0^n 1^m 0^m 1^n \mid n, m \in \mathbb{N}_0\} \ .$$

Beispiele: $001011 \in L_1$, $1100 \in L_1$, $0101 \in L_1$.

(a1) Zeigen Sie, dass L_1 kontextfrei ist, indem Sie eine kontextfreie Grammatik G angeben mit $L(G) = L_1$, und (a2) begründen Sie, warum Ihre Grammatik *genau* die Sprache L_1 erzeugt.

b) Formulieren Sie das Pumping-Lemma für reguläre Sprachen:

„Sei L eine reguläre Sprache über dem Alphabet Σ. Dann gibt es ...“

c) Sei $L_2 = \{0^n 1^m 0^k \mid n, m, k \in \mathbb{N}_0, n \geq 1, m \geq 1, k \geq 1\}$. Zeigen Sie, dass L_2 die Eigenschaft des Pumping-Lemmas für reguläre Sprachen (vgl. (b)) besitzt.

d) Zeigen Sie mit Hilfe des Pumping-Lemmas für reguläre Sprachen, dass die Sprache L_1 aus (a) nicht regulär ist.

Aufgabe 2:

Im Nachfolgenden bezeichne M immer eine deterministische Turingmaschine, die als Eingabe eine natürliche Zahl bekommt, und $\langle M \rangle$ sei Gödelnummer von M.

a) Zeigen Sie, dass die Menge

$$L_1 = \{\langle M \rangle \mid \text{es gibt } \textbf{mindestens} \text{ eine Zahl } n, \text{ so dass } M \text{ gestartet mit } n \text{ hält}\}$$

rekursiv aufzählbar (= partiell-entscheidbar) ist.

Geben Sie dazu einen Algorithmus A an, der als Eingabe eine Gödelnummer $\langle M \rangle$ bekommt und damit gestartet genau dann hält, wenn $\langle M \rangle \in L_1$ ist.

b) Sei Co-$H_0 = \{\langle M \rangle \mid M \text{ gestartet mit } 0 \text{ hält nicht}\}$. Es ist bekannt, dass Co-H_0 nicht rekursiv aufzählbar (= partiell-entscheidbar) ist.

Zeigen Sie durch Reduktion von Co-H_0, dass die Menge

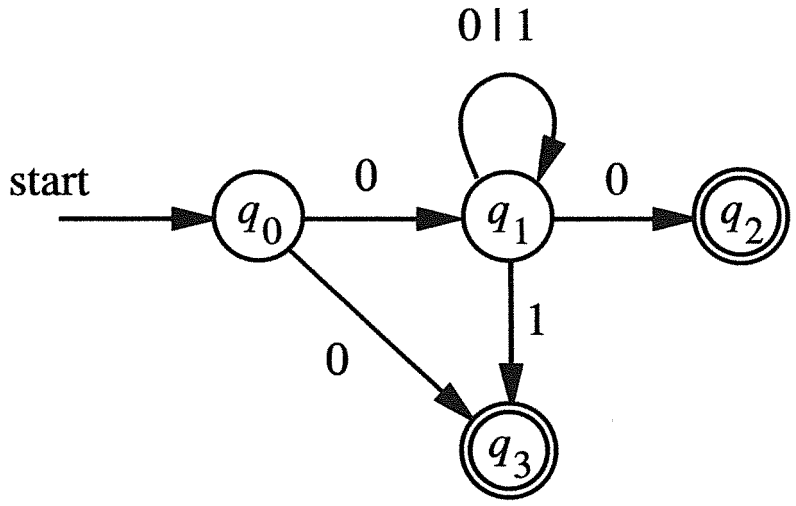
$$L_2 = \{\langle M \rangle \mid \text{es gibt } \textbf{genau} \text{ eine Zahl } n, \text{ so dass } M \text{ gestartet mit } n \text{ hält}\}$$

nicht rekursiv aufzählbar (= partiell-entscheidbar) ist. Zeigen Sie also konkret: Co-$H_0 \leq L_2$.

Fortsetzung nächste Seite!

Aufgabe 3:

a) Gegeben sei der folgende nichtdeterministische endliche Automat N:



Konstruieren Sie zu N mit der Potenzmengen-Konstruktion einen äquivalenten deterministischen endlichen Automaten A. Zeichnen Sie nur die vom Startzustand aus erreichbaren Zustände ein, die aber *alle*. Die Zustandsnamen von A müssen erkennen lassen, wie sie zustande gekommen sind. Führen Sie *keine* „Vereinfachungen" durch!

Hinweis: In einem deterministischen endlichen Automaten muss es an jedem Zustand für jedes Zeichen einen Übergang geben.

b) Geben Sie einen regulären Ausdruck $\alpha(N)$ für die Sprache an, die der nichtdeterministische endliche Automat N aus (a) akzeptiert.

c) Beweisen oder widerlegen Sie die folgende Behauptung.

Beh.: Es gibt reguläre Sprachen L, so dass es eine Teilmenge U von L gibt, die nicht regulär ist.

Aufgabe 4:

a) Gegeben sei die folgende kontextfreie Grammatik $G = (V, \Sigma, P, S)$ mit $V = \{S, A, B, C\}$, $\Sigma = \{a, b\}$ und den Produktionen:

$$S \to A \mid B \mid a \qquad A \to C \mid aBb \qquad B \to Aa \mid b \qquad C \to S \mid bA$$

Konstruieren Sie eine kontextfreie Grammatik G', in der es keine Ketten-Produktionen mehr gibt, mit $L(G') = L(G)$.

Hinweis: Eine Kettenproduktion ist eine Produktion der Form $A \to B$ für $A, B \in V$.

b) Sei $G = (V, \Sigma, P, S)$ eine kontextfreie Grammatik in Chomsky-Normalform, und sei $w = w_1 \dots w_n$ ein Wort aus n Zeichen aus dem Alphabet Σ. Der Algorithmus von Cocke/Younger/Kasami (CYK-Algorithmus) berechnet für alle $i, j \in \{1, \dots, n\}$, $i \leq j$, die Variablenmenge $V(i,j) = \{A \in V \mid A \xrightarrow{*} w_i \dots w_j\}$.

Fortsetzung nächste Seite!

Sei $G = (V, \Sigma, P, S)$ die kontextfreie Grammatik in Chomsky-Normalform mit $V = \{S, A, B, C\}$, $\Sigma = \{a, b\}$ und den folgenden Produktionen:

$$S \to AB \mid BC \qquad B \to CC \mid b$$

$$A \to BA \mid a \qquad C \to AB \mid a$$

Sei $w = baaba$.

Folgende Tabelle entsteht durch Anwendung des CYK-Algorithmus. Z. B. bedeutet $B \in V(3,5)$, dass aus der Variablen B das Teilwort $w_3w_4w_5 = aba$ hergeleitet werden kann.

Tragen Sie in der Tabelle die drei fehlenden Einträge $V(1,2)$, $V(1,4)$ und $V(1,5)$ ein. Wie entnehmen Sie dieser Tabelle, dass $w \in L(G)$ ist?

b	a	a	b	a
$V(1,1)$ $\{B\}$	$V(2,2)$ $\{A,C\}$	$V(3,3)$ $\{A,C\}$	$V(4,4)$ $\{B\}$	$V(5,5)$ $\{A,C\}$
$V(1,2)$	$V(2,3)$ $\{B\}$	$V(3,4)$ $\{S,C\}$	$V(4,5)$ $\{S,A\}$	
$V(1,3)$ $\varnothing$	$V(2,4)$ $\{B\}$	$V(3,5)$ $\{B\}$		
$V(1,4)$	$V(2,5)$ $\{S,A,C\}$			
$V(1,5)$				