
Prüfungsteilnehmer	Prüfungstermin	Einzelprüfungsnummer
--------------------	----------------	----------------------

Kennzahl: _____

Kennwort: _____

Arbeitsplatz-Nr.: _____

Frühjahr
2009

66115

Erste Staatsprüfung für ein Lehramt an öffentlichen Schulen
— Prüfungsaufgaben —

Fach: **Informatik (vertieft studiert)**

Einzelprüfung: **Theoretische Informatik, Algorithmen**

Anzahl der gestellten Themen (Aufgaben): 2

Anzahl der Druckseiten dieser Vorlage: 9

Bitte wenden!

Thema Nr. 1
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1 Ordnen Sie die folgenden formalen Sprachen bestmöglich in die Chomsky-Hierarchie ein, und geben Sie eine ausreichende Begründung dafür, dass die jeweilige Sprache in der von Ihnen genannten Hierarchiestufe liegt und in keiner niedrigeren eingeordnet werden kann!

1. $L_1 = \{ a^n b^n a^n \mid n \geq 1 \}$
2. $L_2 = \{ a^n b^m a^n \mid m \geq 1, n \geq 1 \}$
3. $L_3 = \{ a^n b^m \mid m \geq 1, n \geq 1 \}$
4. $L_4 =$ Die Menge aller Dezimaldarstellungen von Vielfachen von 3
5. $L_5 =$ Die Menge aller terminierenden Java-Programme
6. $L_6 = \{ w \in \{a, b\}^* \mid w \text{ enthält mindestens 4 Vorkommen von } a \}$.

Aufgabe 2 Wir betrachten den deterministischen Automaten $(\Sigma, Z, z_0, \delta, F)$, wobei $\Sigma = \{0, 1\}$ und $Z = \{\epsilon, 0, 1, 00, 01, 10, 11\}$ und $z_0 = \epsilon$ und $F = \{10\}$ und die Zustandsüberföhrungsfunktion δ schematisch durch die folgenden Gleichungen gegeben ist:

$$\begin{aligned}\delta(\epsilon, x) &= x \\ \delta(x, y) &= xy \\ \delta(xy, z) &= yz\end{aligned}$$

wobei $x, y, z \in \{0, 1\}$. Es gilt also zum Beispiel $\delta(0, 1) = 01$ und $\delta(01, 1) = 11$. Lassen Sie sich nicht dadurch verwirren, dass die Zustände hier Wörter sind.

1. Zeichnen Sie den Automaten als Diagramm!
2. Geben Sie einen möglichst kurzen regulären Ausdruck für die erkannte Sprache L an!
3. Wenden Sie den Minimierungsalgorithmus auf den Automaten an und berechnen Sie so den Minimalautomaten!
4. Geben Sie die Äquivalenzklassen der Myhill-Nerode Äquivalenzrelation als reguläre Ausdrücke an! Hinweis: Die Myhill-Nerode Äquivalenz ist erklärt durch $u \sim v \iff (\forall w : uw \in L \iff vw \in L)$.

Tipp: Die Vereinigung aller Klassen muss $\{0, 1\}^*$ ergeben.

Aufgabe 3 Welche der folgenden Aussagen trifft zu, welche nicht? Begründen Sie jeweils Ihre Antwort durch Beweis oder Gegenbeispiel! Natürlich dürfen Sie sich auf allgemein bekannte Lehrsätze und den Informatikduden beziehen.

1. Ist eine Sprache L rekursiv aufzählbar, so ist sie auch entscheidbar.
2. Ist eine Sprache auf das Halteproblem reduzierbar, so ist sie rekursiv aufzählbar.
3. Die rekursiv aufzählbaren Sprachen sind unter Vereinigung und Durchschnitt abgeschlossen.
4. Die rekursiv aufzählbaren Sprachen sind unter Komplement abgeschlossen.
5. Es ist unentscheidbar, ob eine Turingmaschine M bei leerer Eingabe im Verlaufe der Berechnung eine Konfiguration erreicht, die grösser als 17 ist. Tipp: Es gibt nur eine feste Zahl von Konfigurationen der Grösse 17 oder kleiner.

Aufgabe 4 Hier finden Sie Pseudocode¹ für den Dijkstra-Algorithmus in der Notation des Informatikdudens. Insbesondere ist R eine Prioritätsschlange mit $D[]$ als Schlüssel. Der Algorithmus berechnet die kürzesten Entfernungen vom Startknoten s in einem gerichteten Graphen $G = (V, E)$ mit nichtnegativen Kantengewichten $d(\cdot, \cdot)$. Der Aufruf $\text{deletemin}(R, v)$ weist den kleinsten Eintrag von R der Variablen v zu und entfernt ihn aus R .

```

setze  $D[v] = \infty$  für alle Knoten  $v$ ;
 $D[s] := 0$ ;  $R := V$ 
while  $R \neq \emptyset$  do  $\text{deletemin}(R, v)$ ;
  for all Knoten  $w$  mit  $(v, w) \in E$  do
     $h := D[v] + d(v, w)$ ;
    if  $h < D[w]$ 
      then  $D[w] := h$ ;
      decreasekey( $R, w$ );

```

1. Arbeiten Sie den Algorithmus für folgende Eingabe ab und dokumentieren Sie die Schritte geeignet!

$V = \{s, u, v, x, y\}$
 $E = \{(s, u), (s, x), (u, x), (x, u), (u, v), (x, y), (x, v), (v, y), (y, v), (y, s)\}$
 $d(s, u) = 10, d(s, x) = 5, d(u, x) = 2, d(x, u) = 3, d(u, v) = 1,$
 $d(x, y) = 2, d(x, v) = 9, d(v, y) = 4, d(y, v) = 6, d(y, s) = 7$

2. Der syldawische Informatiker W. E. ARTSKJID schlug 2007 vor, beim Abarbeiten eines Knotens v jeweils die Menge $M = \{v_1, \dots, v_k\}$ derjenigen Knoten in R zu bilden, deren D -Wert während der Abarbeitung von v verringert wurde. Aus dieser Menge M wird nun der Knoten mit dem kleinsten D -Wert gesucht und als nächstes abgearbeitet. Sollte die Menge M leer sein, so wird wie üblich verfahren, es wird also der Knoten aus R mit kleinstem D -Wert als nächstes abgearbeitet.

Geben Sie Pseudocode für den so modifizierten Algorithmus an!

Geben Sie ein konkretes Beispiel dafür, dass der modifizierte Algorithmus bisweilen unkorrekte Ergebnisse liefert! Sie müssen dazu den entsprechenden Eingabegraphen mit Kantengewichten skizzieren und den Lauf des modifizierten Algorithmus auf dieser Eingabe geeignet dokumentieren.

¹Die Version der Informatikdudens enthält eine hier störende Ungenauigkeit, welche es im Prinzip erlaubt, bereits abgearbeitete Knoten wieder in die Prioritätsschlange aufzunehmen.

Thema Nr. 2
(Aufgabengruppe)

Es sind alle Aufgaben dieser Aufgabengruppe zu bearbeiten!

Aufgabe 1:

Formale Sprachen und Automatentheorie

- a) Wir betrachten die Sprache L über der Zeichenmenge $T = \{a, b\}$, die das Wort $abab$ als Teilwort enthalten.
- i) Geben Sie einen regulären Ausdruck an, der L beschreibt.
 - ii) Geben Sie eine reguläre, rechtslineare Grammatik an, die L erzeugt.
 - iii) Geben Sie eine Ableitung des Wortes $aababbb$ an.
- b) Gegeben sei der reguläre Ausdruck:

$$[0|1[01^*0]^*1]^*.$$

Wandeln Sie diesen regulären Ausdruck in einen nichtdeterministischen endlichen Automaten um.

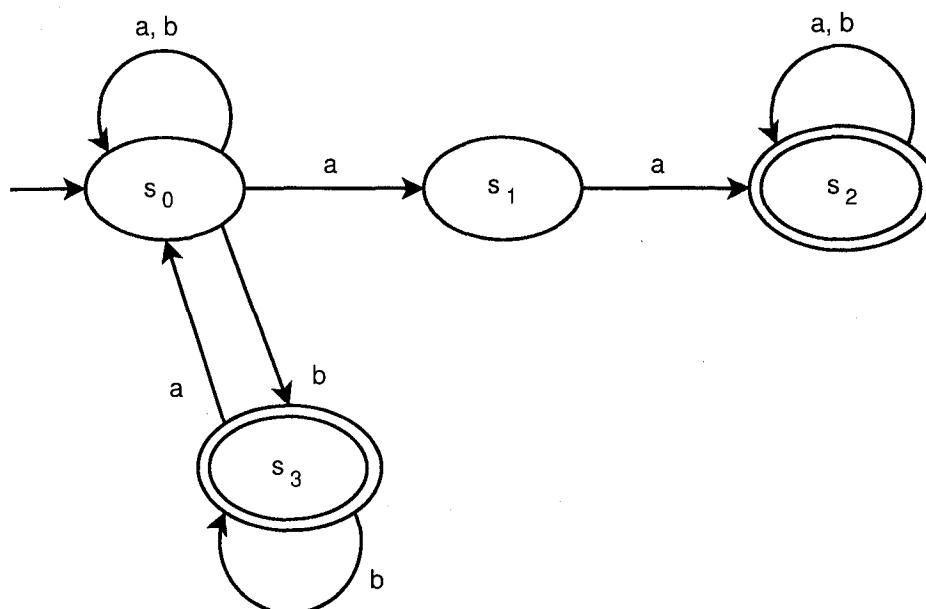
- c) Gegeben sei der endliche, nichtdeterministische Automat

$$A = (S, T, s_0, \delta) \text{ mit}$$

$$S = \{s_0, s_1, s_2, s_3\},$$

$$T = \{a, b\}, Z = \{s_2, s_3\}$$

mit der Übergangsfunktion δ gemäß untenstehendem Diagram.



- i) Konstruieren Sie einen äquivalenten, deterministischen, endlichen Automaten.
- ii) Minimalisieren Sie den endlichen Automaten.

Fortsetzung nächste Seite!

- d) Eine Turing-Maschine D , die natürliche Zahlen in Strichdarstellung halbiert (ganzzahlige Division durch 2), mit Eingabealphabet $T = \{ | \}$ soll in mehreren Schritten entwickelt werden.
- Definieren Sie eine Turing-Maschine L mit Eingabealphabet T , die den Kopf zum nächsten leeren Bandfeld nach links (mindestens um ein Feld) bewegt und die Bandbeschriftung nicht ändert. L setzt voraus, dass der Kopf auf einem leeren Bandfeld steht (Beispiel für Startkonfiguration: $\# ||| \underline{\#}$).
- Stellen Sie die Übergangsrelation von L in Form einer Tabelle und in Form eines erweiterten, endlichen Automaten dar.
- Hinweis:** Ein erweiterter, endlicher Automat ist ein endlicher Automat, dessen Zustandsübergänge neben der gelesenen Eingabe mit der geschriebenen Ausgabe und der Lese-/Schreibkopfbewegung annotiert sind.
- Definieren Sie analog zu i) eine Turing-Maschine R , die den Kopf zum nächsten leeren Bandfeld nach rechts bewegt. Geben Sie eine vollständige Berechnung von R zur Startkonfiguration $\underline{\#} ||| \#$ an.
 - Geben Sie nun unter Verwendung von L und R eine Turing-Maschine D (für die ganzzahlige Division durch 2 einer Strichzahl) in Form eines erweiterten, endlichen Automaten an. D setzt voraus, dass der Kopf auf dem ersten leeren Bandfeld rechts von der Eingabe steht. Beispielberechnung: $(\### ||| \underline{\#}) \rightarrow_* (\# | \# \underline{\#} \# \# \#)$

Aufgabe 2:

Berechenbarkeit

- Gegeben sei die Funktion $f : \mathbb{N}_0 \rightarrow \mathbb{N}_0$ mit $f(x) = x - 1$ wobei $'-'$ definiert ist als Subtraktion über den natürlichen Zahlen mit der Besonderheit, dass $a - b = 0$ ergibt, falls $a \leq b$.
 - Zeigen Sie, dass die Funktion f Einband-Turing-berechenbar ist, indem Sie eine solche Turing-Maschine angeben. x sei dabei wieder in Strichdarstellung gegeben. Der Kopf beginnt rechts.
 - Nennen Sie zwei weitere Modelle für Berechenbarkeit.
 - Was besagt die Church These im Hinblick auf die obigen Modelle?
- Begründen Sie, dass die Menge der Primzahlen entscheidbar ist.
 - Zeigen Sie folgende Eigenschaften für entscheidbare Mengen:
 - Jede endliche Menge, einschließlich der leeren Menge, ist entscheidbar.
 - Das Komplement einer entscheidbaren Menge ist entscheidbar.
- Sei $A \subseteq \mathbb{N}_0$.
 - Geben Sie die Definition für rekursive Aufzählbarkeit dieser Menge an.
 - Zeigen Sie, dass eine Menge A genau dann entscheidbar ist, wenn A und \overline{A} rekursiv aufzählbar sind.

Aufgabe 3:

Komplexität

- a) Die folgenden Fragen beziehen sich auf nichtdeterministische Turing-Maschinen.
- Geben Sie eine Definition in üblicher Tupelschreibweise an.
 - Erarbeiten Sie die wesentlichen Unterschiede zu einer deterministischen Turing-Maschine.
 - Was bedeutet die Klasse NP ?
- b) Gegeben sei das SOS (Sum of Subset) - Problem

$$SOS := \left\{ (a_1, \dots, a_m, b) : m, a_1, \dots, a_m, b \in \mathbb{N}_0 \right. \\ \left. \text{und es existieren } c_1, \dots, c_m \in \{0, 1\} \text{ mit} \right. \\ \left. \sum_{i=1}^m c_i \cdot a_i = b \right\}$$

- Zeigen Sie, dass das SOS - Problem in polynomialer Zeit auf einer nichtdeterministischen Maschine entscheidbar ist (verwenden Sie dazu ein geeignetes übliches nichtdeterministisches Berechnungsmodell in dem Arrays als primitive Datenstruktur vorhanden sind).
- Gegeben sei das NP-vollständige Problem TS' , eine Variante des "Travelling Salesperson" Problems:

$$TS' := \left\{ (m, M, k) : m, k \in \mathbb{N}_0, M : \{1, 2, \dots, m\}^2 \rightarrow \mathbb{N}_0 \right. \\ \left. \text{und es existiert eine Permutation } \pi \text{ mit} \right. \\ \left. \sum_{i=1}^{m-1} M(\pi(i), \pi(i+1)) + M(\pi(m), \pi(1)) = k \right\}$$

Zeigen Sie damit, dass das obige SOS-Problem ebenfalls ein NP-vollständiges Problem ist.

Aufgabe 4:

Effiziente Algorithmen und Datenstrukturen

- a) Die Ausdrücke $O(\cdot)$ und $\Omega(\cdot)$ sind gebräuchliche Notation in der Informatik.
- Wie sind sie jeweils definiert?
 - Benennen Sie die wesentlichen Unterschiede.

- iii) Geben Sie in $O(\cdot)$ -Notation die Berechnungskomplexität in Abhängigkeit von a und b für folgenden Algorithmus an:

```

function fastmult(a,b);
begin
  A[0] := 1;
  B[0] := a;
  j := 0;
  while (A[j] <= b) do begin
    A[j+1] := (A[j] + A[j]);
    B[j+1] := (B[j] + B[j]);
    j := (j +1)
  end;
  c := 0;
  while (j > 0) do begin
    j := (j-1);
    if (A[j] <= b) then begin
      b := (b - A[j]);
      c := (c + B[j])
    end
  end;
  fastmult := c;
end

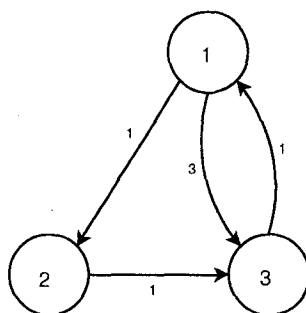
```

- iv) Zeigen oder widerlegen Sie:

$$2^{n+1} = O(2^n)?$$

$$2^{2n} = O(2^n)?$$

- b) Gegeben sei der Graph G :



Somit ist die Adjazenzmatrix von G die Matrix D_0 :

$$D_0 = \begin{pmatrix} \infty & 1 & 3 \\ \infty & \infty & 1 \\ 1 & \infty & \infty \end{pmatrix}$$

Fortsetzung nächste Seite!

Um den kürzesten Pfad in diesem gewichteten, gerichteten Graphen zu finden, setzen wir den Floyd-Warshall-Algorithmus ein:

```
 $n := \text{rows}[D_0]$ 

for  $k \leftarrow 1$  to  $n$  do

    for  $i \leftarrow 1$  to  $n$  do

        for  $j \leftarrow 1$  to  $n$  do

             $D_k[i][j] \leftarrow \min(D_{k-1}[i][j], D_{k-1}[i][k] + D_{k-1}[k][j])$ 

return  $D_n$ 
```

- i) Führen Sie den Floyd-Warshall Algorithmus auf der den Graphen G repräsentierenden Matrix D_0 aus. Stellen Sie dabei die Zwischenergebnisse der Matrizen D_1 und D_2 sowie das Ergebnis D_3 in Matrixform dar.
- ii) Auf welchen gewichteten, gerichteten Graphen berechnet der Floyd-Warshall-Algorithmus genau die transitive Hülle?