Dr. Rainer Feldmann Paderborn, den 19.2.2007

2. Klausur

in

Einführung in Berechenbarkeit, formale Sprachen und Komplexitätstheorie

WS 2006/2007

Name : ....

Matrikelnummer : ....

Punkteverteilung (bitte freilassen!)

Formale Sprachen		
Aufgabe 1		von 20
Aufgabe 2		von 20
Aufgabe 3		von 20
Summe		von 60

Berechenbarkeit+Komplexität		
Aufgabe 4		von 20
Aufgabe 5		von 20
Aufgabe 6		von 20
Summe		von 60

Gesamtsumme		von 120

Die Klausur besteht aus 6 Aufgaben und 22 Seiten. Es können insgesamt 120 Punkte erreicht werden. Bitte schreiben Sie auf jedes Blatt Ihre Matrikelnummer.

Teilnehmer, die nur den Teil “Berechenbarkeit+Komplexität” abdecken wollen, bearbeiten bitte nur die Aufgaben 4,5,6. Die Bearbeitungszeit beträgt 90 Minuten.

Die Seiten 21-22 sind Konzeptpapier und werden nicht bewertet.

Zugelassene Hilfsmittel: Ein Blatt DIN A4, handschriftlich beidseitig beschrieben.

Schreiben Sie Ihre Lösung bitte an den dafür vorgesehenen Platz. Sollte der Platz nicht ausreichen, erhalten Sie auf Anfrage weiteres Papier.

Aufgabe 1 [FS:NEA-DEA] (20 Punkte)

Gegeben ist ein NEA $M = (Q, \Sigma, \delta, q_0, F)$. Gesucht ist ein zu M äquivalenter DEA $\tilde{M} = (\tilde{Q}, \tilde{\Sigma}, \tilde{\delta}, \tilde{q_0}, \tilde{F})$.

a) Geben Sie $\tilde{Q}, \tilde{\Sigma}, \tilde{q_0}, \tilde{F}$ formal an:

$\tilde{Q} =$

$\tilde{\Sigma} =$

$\tilde{q_0} =$

$\tilde{F} =$

b) Gegeben ist ein NEA $M = (Q, \Sigma, \delta, q_0, F)$ mit $Q = \{a, b, c, d\}$, $\Sigma = \{0, 1, 2\}$, $q_0 = a$, $F = \{d\}$ und δ wie in nachfolgender Tabelle:

δ	0	1	2
a	$\{a, b\}$	$\{a\}$	$\{a\}$
b	$\emptyset$	$\{c\}$	$\emptyset$
c	$\emptyset$	$\emptyset$	$\{d\}$
d	$\{d\}$	$\{d\}$	$\{d\}$

Geben Sie $\tilde{\delta}$ in der Tabelle auf der nächsten Seite an. Beschränken Sie sich dabei auf die Angabe der erreichbaren Zustände von $\tilde{M}$ und markieren Sie finale Zustände mit $''X''$ in der rechten Spalte.

$\tilde{\delta}$	0	1	2	final

Tabelle für 2. Versuch:

$\tilde{\delta}$	0	1	2	final

Aufgabe 2 [FS:kontextfreie Grammatik] (20 Punkte)

Die *umgekehrt polnische Notation (UPN)* für arithmetische Ausdrücke über $\mathbb{N}_0$ ist eine Notation, die arithmetische Ausdrücke darstellen kann und keine Klammern benötigt. Die Operatoren (hier $\{+, -, *\}$) werden dabei hinter die Operanden (hier: $\in \mathbb{N}_0$, jeweils mit '.' beendet) geschrieben. Also zum

Beispiel

	normal	**UPN**
a)	1	$1.$
b)	$1+2$	$1.2.+$
c)	$(1-2)*3$	$1.2.-3.*$
d)	$3*(1-2)$	$3.1.2.-*$
e)	$(3-1)*(2+5)$	$3.1.-2.5.+*$
f)	$(99-10)*((133+16)*3)$	$99.10.-133.16.+3.**$

a) Geben Sie formal und vollständig eine kontextfreie Grammatik an, die die folgende Sprache über dem Alphabet $\Sigma = \{0, \dots, 9, +, -, *, .\}$ erzeugt:

$U = \{w \in \Sigma \mid w$ ist korrekter arithm. Ausdruck über $\mathbb{N}_0$ und $\{+, -, *\}$ in UPN$\}$.

b) Geben Sie für das Wort $99.10. - 133.16. + 3. **$ aus Beispiel f) oben eine Ableitung in Ihrer Grammatik an.

c) Beweisen Sie durch Induktion über $|w|$ für Ihre Grammatik G: Ist $w \in L(G)$, so ist $w \in U$.

Zusätzlicher Platz für die Lösung der Aufgabe 2:

Zusätzlicher Platz für die Lösung der Aufgabe 2:

Zusätzlicher Platz für die Lösung der Aufgabe 2:

Aufgabe 3 [FS:Pumping Lemma] (20 Punkte)

Beweisen Sie explizit durch Anwendung von Abschlusseigenschaften und Anwendung des Pumping Lemmas für kontextfreie Sprachen, dass die Sprache

$$L = \{wc^k w \mid w \in \{a,b\}^*,\ k \in \mathbb{N}\}$$

nicht kontexfrei ist.

Zusätzlicher Platz für die Lösung der Aufgabe 3:

Aufgabe 4 [BK:NTM] (20 Punkte)

a) Geben Sie formal und vollständig eine nichtdeterministische 2-Band Turingmaschine an, die die Sprache

$$L = \{1^k \mid k \geq 2 \text{ ist keine Primzahl}\}$$

akzeptiert.

b) Kommentieren Sie in der Tabelle rechts jeden Zustand q ihrer NTM ausgiebig, indem Sie den Inhalt der beiden Bänder und die Kopfpositionen beschreiben, wenn ihre Maschine in den Zustand q geht. Für den Startzustand q_0 ist das bereits beispielhaft angegeben.

Zustand	Kommentar
q_0	Auf Band 1 steht genau die Eingabe $1 \ldots 1$ beginnend auf Bandposition 1, auf Band 2 stehen nur Blanks. Kopf 1 steht auf der linkesten 1, Kopf 2 steht an Position 1 von Band 2.

Zusätzlicher Platz für die Lösung der Aufgabe 4:

Zusätzlicher Platz für die Lösung der Aufgabe 4:

Aufgabe 5 [BK:Reduktion] (20 Punkte)

Beweisen Sie durch Reduktion, dass das folgende Problem unentscheidbar ist:

Problem TM-REG-ÄQUIVALENZ**:**

geg.: Eine Turingmaschine M (gegeben als Kodierung $< M >$) und eine reguläre Grammatik G (gegeben als Kodierung $< G >$).
Frage: Gilt L(M) = L(G) ?

Hinweis: Nutzen Sie die Mengen $K := \{< M >< G > \mid L(M) = L(G)\}$ und $L = \{< M > \mid M$ akzeptiert keine Eingabe$\}$.

Zusätzlicher Platz für die Lösung der Aufgabe 5:

Aufgabe 6 [BK:Approximation] (20 Punkte)

Das RUCKSACK Problem ist wie folgt formal definiert:

Problem RUCKSACK**:**

> **geg.:** $n \in \mathbb{N}$, $u \in \mathbb{N}^n$, $w \in \mathbb{N}^n$, $c \in \mathbb{N}$.
> **ges.:** $I \subseteq [n]$, so dass $\sum_{i \in I} w_i \leq c$ und $\sum_{i \in I} u_i$ maximal ist.

Betrachten Sie den folgenden Algorithmus für Rucksack:

BADKNAPSACK(n, u, w, c) $\{$
  sortiere die n Gegenstände so, dass die Folge $(\frac{u_i}{w_i})_{i=1,\ldots,n}$ monoton fallend ist;
  /* es gelte jetzt: $\frac{u_{i+1}}{w_{i+1}} \leq \frac{u_i}{w_i}$ für alle $i \in [n-1]$ */
  $I = \emptyset$;
  for $i = 1$ **to** n $\{$
    if $w_i \leq c$ $\{$
      $I = I \cup \{i\}$; $c = c - w_i$;
    $\}$
  $\}$ /* for ... */
  return(I);
$\}$

a) Vervollständigen Sie die Instanz $J(n,k) = (n,u,w,c)$ wie in der Tabelle unten angegeben, so daß die Güte $R(n,k)$ von BADKNAPSACK aus $J(n,k)$ größer ist als k und beweisen Sie die Aussage $R(n,k) > k$ für alle $n, k \geq 2$ formal.

$n \in \mathbb{N}$ Gegenstände,
$c = k \cdot n$ mit einer beliebig großen Konstante $k \in \mathbb{N}$ und

i	1	$\ldots$	$n-1$	n
u_i	1	$\ldots$	1	
w_i	1	$\ldots$	1	

b) Beschreiben Sie, warum das Verhalten von BADKNAPSACK auf den Instanzen $J(n,k)$ aus a) so schlecht ist.

Teil c) der Aufgabe auf der nächsten Seite!

c) Geben Sie einen Algorithmus GOODKNAPSACK an, der BADKNAPSACK als Unterprogramm verwenden darf und

1) niemals eine schlechtere Lösung liefert als BADKNAPSACK und
2) auf allen Instanzen $J(n, k)$ eine optimale Lösung liefert und
3) auf mindestens einer Instanz $J \notin \{J(n, k) \mid n, k \in \mathbb{N}\}$ eine bessere Lösung als BADKNAPSACK liefert und
4) asymptotisch die gleiche Laufzeit hat wie BADKNAPSACK.

Begründen Sie die Aussagen 1) bis 4) für Ihren Algorithmus.

Zusätzlicher Platz für die Lösung der Aufgabe 6:

Zusätzlicher Platz für die Lösung der Aufgabe 6:

Konzeptpapier (wird nicht bewertet)

Konzeptpapier (wird nicht bewertet)