Hochschuldozent Dr. Christian Schindelhauer Paderborn, den 21. 2. 2006

Lösungen zur 1. Klausur

in

Einführung in Berechenbarkeit, formale Sprachen und Komplexitätstheorie

Name : ....

Matrikelnummer : ....

Punkteverteilung (bitte freilassen!)

Aufgabe 1		von 200
Aufgabe 2		von 200
Aufgabe 3		von 250
Aufgabe 4		von 250
Aufgabe 5		von 100
Summe		von 1000

Die Klausur besteht aus fünf Aufgaben und 18 Seiten. Bitte schreiben Sie auf jedes Blatt Ihre Matrikelnummer.

Zugelassene Hilfsmittel: Ein eigenhändig beidseitig handbeschriebenes DIN A4-Blatt.

Schreiben Sie Ihre Lösung bitte in die vorgesehenen Platzhalter. Sollte der Platz nicht ausreichen, erhalten Sie auf Anfrage weiteres Papier.

Aufgabe 1 (200 Punkte)

a) (100 Punkte)

Gegeben ist folgende Sprache:

$$L := \{x \in \{0,1\}^* \mid x \text{ enthält } 110 \text{ und endet auf } 0\}$$

Ordnen Sie diese Sprache in die höchstmögliche Klasse der Chomsky-Hierarchie ein. Kreuzen Sie dazu den entsprechenden Typ an. Zeigen Sie, dass L in genau dieser Klasse ist und zeigen Sie, warum L nicht höher eingeordnet werden kann.

- ☐ Chomsky-0 (rekursiv aufzählbare Sprachen)
- ☐ Chomsky-1 (kontextsensitive Sprachen)
- ☐ Chomsky-2 (kontextfrei Sprachen)
- ☐ Chomsky-3 (reguläre Sprachen)

Fortsetzung von Aufgabe 1 a):

b) (100 Punkte)

Gegeben ist folgende Sprache:

$$L := \{a^i b^j c^k \mid i, j, k \geq 1, i = j \text{ oder } i = k \text{ oder } j = k\}$$

Ordnen Sie diese Sprache in die höchstmögliche Klasse der Chomsky-Hierarchie ein. Kreuzen Sie dazu den entsprechenden Typ an. Zeigen Sie, dass L in genau dieser Klasse ist und zeigen Sie, warum L nicht höher eingeordnet werden kann.

- ☐ Chomsky-0 (rekursiv aufzählbare Sprachen)
- ☐ Chomsky-1 (kontextsensitive Sprachen)
- ☐ Chomsky-2 (kontextfrei Sprachen)
- ☐ Chomsky-3 (reguläre Sprachen)

Fortsetzung von Aufgabe 1 b):

Aufgabe 2 (200 Punkte)

Betrachten Sie die folgende Grammatik $G = (V, \Sigma, S, P)$. Es ist $V = \{S, U, W, X, Y, Z\}$, $\Sigma = \{x, y\}$ und die Menge P der Produktionen ist gegeben durch:

$$\begin{array}{rcl@{\qquad}rcl}
S & \rightarrow & XY \mid YZ \mid \epsilon & U & \rightarrow & x \\
X & \rightarrow & YX & Y & \rightarrow & y \\
Y & \rightarrow & ZZ & Z & \rightarrow & x \\
Z & \rightarrow & XY & & & \\
X & \rightarrow & W & & & \\
W & \rightarrow & U & & &
\end{array}$$

a) (50 Punkte)

Formen Sie die Grammatik G zunächst in Chomsky-Normalform um.

b) (100 Punkte)

Entscheiden Sie nun mit Hilfe des CYK-Algorithmus, ob das Wort $yxxyx$ in $L(G)$ liegt. Geben Sie hierfür die vom CYK-Algorithmus erzeugte Tabelle $T(i, j)$ an.

$l = 5$					
$l = 4$					
$l = 3$					
$l = 2$					
$l = 1$					

Platz für einen zweiten Versuch (gültigen Versuch kennzeichnen!):

$l = 5$					
$l = 4$					
$l = 3$					
$l = 2$					
$l = 1$					

Folgerung:

c) (50 Punkte)

Geben Sie alle Teilwörter des Wortes $yxxyx$ an, die in $L(G)$ liegen.

Lösung:

G in Chomsky-Normalform:

$$
\begin{array}{rcl}
S & \rightarrow & XY \mid YZ \mid \epsilon \\
X & \rightarrow & YX \\
Y & \rightarrow & ZZ \\
Z & \rightarrow & XY
\end{array}
\qquad
\begin{array}{rcl}
X & \rightarrow & x \\
Y & \rightarrow & y \\
Z & \rightarrow & x
\end{array}
$$

Tabelle des CYK-Algorithmus:

$l=5$	$\{S,X\}$				
$l=4$	$\{\}$	$\{S,X,Z\}$			
$l=3$	$\{\}$	$\{Y\}$	$\{Y\}$		
$l=2$	$\{X,S\}$	$\{Y\}$	$\{S,Z\}$	$\{X,S\}$	
$l=1$	$\{Y\}$	$\{X,Z\}$	$\{X,Z\}$	$\{Y\}$	$\{X,Z\}$
	y	x	x	y	x

Teilwörter aus $yxxyx$ die in $L(G)$ liegen:

ϵ, yx, xy, $xxyx$

Aufgabe 3 (250 Punkte)

Betrachten Sie die folgende Sprache L:

$$L = \{\langle M\rangle \mid M \text{ akzeptiert mindestens 3 Eingaben } x \text{ mit } |x| = 5\}$$

a) (100 Punkte)

Zeigen Sie, dass L rekursiv aufzählbar ist.

Fortsetzung von Aufgabe 3 a):

Lösung:

a) 2 Band DTM, die Woerter der Laenge 5 in betrachtet
Es gibt nur endlich viele solcher Worte. DTM fuehrt für alle diese Worte zuerst 1 Schritt von M durch, anschliessend 2 Schritte, ...
Falls ein Wort akzeptiert wird, wird dieses auf Band 2 gespeichert
Jedes weitere Wort wird mit gefundenem verglichen, falls neues Wort gefunden wurde, ebenfalls speichern
Bei drittem gefundenem Wort akzeptieren.

b) Lösung für H, für A analog
H ist

- rek. aufz.
- nicht entscheidbar
- nicht rek. Ko-aufz.
- $\notin P$
- $\notin NP$

$\overline{H} \leq_m \overline{L} \Longleftrightarrow H \leq_m L$
O.b.d.A. ist Kodierung so, dass für jedes $w \in \Sigma^*$ gilt, w ist von der Form $w = \langle M, x \rangle$
Reduktionsfunktion: $f(\langle M, x \rangle) = M'$
Arbeitsweise von M':
Lösche Eingabe
Lasse M mit Eingabe x laufen
Wenn M hält akzeptiere

Korrektheit: $\langle M, x \rangle \in HALT \Longrightarrow$ Simulation von $\langle M, x \rangle$ auf M' hält für jede Eingabe, $\Longrightarrow M'$ akzeptiert jede Eingabe, insbesondere also mindestens 3 Eingaben der Länge 5 $\Longrightarrow M' \in L$
$\langle M, x \rangle \notin HALT \Longrightarrow$ Simulation von $\langle M, x \rangle$ auf M' hält für keine Eingabe, $\Longrightarrow M'$ hält für keine Eingabe, insbesondere akzeptiert M' keine Eingabe der Länge 5 $\Longrightarrow M' \notin L$

c) Aus Vorlesung bekannt: ist L nicht entscheidbar, dann ist L oder $\overline{L}$ nicht rekursiv aufzählbar.

Nach a) ist L rekursiv aufz., es genügt also zu zeigen, dass L nicht entscheidbar ist.
Sei $K' := \{S \subseteq \Sigma^* | \exists T \subseteq S$ mit $|x| = 5 \forall x \in T$ und $|T| \geq 3\}$
und K die rekursiv aufz. Sprachen aus K'
Dann ist $L = \{\langle M \rangle | M$ ist DTM und $L(M) \in K\}$
K ist nicht trivial und enthält nicht alle rek. aufz. Sprachen
Nach Satz von Rice ist L nicht entscheidbar

b) (100 Punkte)

Zeigen Sie mit Hilfe einer Abbildungsreduktion, dass L nicht rekursiv ko-aufzählbar ist. Kreuzen Sie zunächst die Sprache an, die Sie für Ihre Reduktion verwenden möchten:

- ☐ $H = \{\langle M, x\rangle \mid M$ hält bei Eingabe $x\}$
- ☐ $A = \{\langle M, x\rangle \mid M$ akzeptiert Eingabe $x\}$
- ☐ $T = \{\langle M\rangle \mid M$ hält für jede Eingabe$\}$

Kreuzen Sie an, welche Aussagen für die von Ihnen gewählte Sprache gelten:

- Die Sprache ist rekursiv aufzählbar.
  ☐ Richtig ☐ Falsch
- Die Sprache ist entscheidbar.
  ☐ Richtig ☐ Falsch
- Die Sprache ist rekursiv ko-aufzählbar.
  ☐ Richtig ☐ Falsch
- Die Sprache liegt in $\mathcal{P}$.
  ☐ Richtig ☐ Falsch
- Die Sprache liegt in $\mathcal{NP}$.
  ☐ Richtig ☐ Falsch

Führen Sie nun die Reduktion durch:

Fortsetzung von Aufgabe 3 b):

c) (50 Punkte)

Könnte Aufgabenteil b) unter Zuhilfenahme der Ergebnisse aus Teil a) auch mit Hilfe des Satzes von Rice gelöst werden ? Begründen Sie Ihre Antwort.

Aufgabe 4 (250 Punkte)

Seien $G = (V_1, E_1)$ und $H = (V_2, E_2)$ zwei ungerichtete Graphen. G besitzt einen *zu H isomorphen Subgraph*, wenn die graphische Darstellung von H sich passend über einen Teil von G legen lässt. Dies wird durch eine bijektive Abbildung modelliert, die den entsprechenden Teil der Knoten $V \subseteq V_1$ auf die Knoten V_2 von H abbildet.

Formal: G besitzt einen zu H isomorphen Subgraph, wenn es Teilmengen $V \subseteq V_1$ und $E \subseteq E_1$ und eine bijektive Abbildung $f : V \mapsto V_2$ gibt, so dass $|V| = |V_2|$, $|E| = |E_2|$ und $\{u, v\} \in E \Leftrightarrow \{f(u), f(v)\} \in E_2$.

Die Sprache SUBGRAPH ist nun wie folgt definiert:

$$\textsc{Subgraph} := \left\{ \langle G, H \rangle \;\middle|\; \begin{array}{l} G \text{ und } H \text{ sind ungerichtete Graphen } \textit{und} \\ G \text{ enthält einen zu } H \text{ isomorphen Subgraphen} \end{array} \right\}$$

a) (100 Punkte)

Zeigen Sie, dass die Sprache SUBGRAPH in $\mathcal{NP}$ liegt.

Beweis: durch Angabe eines Verifizierers. Im folgenden kodiere das übergebene Zertifikat c die geforderte Funktion f in geeigenter Darstellung:

Algorithm 1 SUBGRAPHVERIFIER

procedure SUBGRAPHVERIFIER($G, H, c = (V, E)$)
    Ist $|V| = |V_2|$ und $|E| = |E_2|$ fahre fort, sonst breche ab
    for $e = (u, v) \in E$ **do**
        if $(f(u), f(v)) \notin E_2$ **then** breche ab
        end if
    end for
    Akzeptiere
end procedure

Korrektheit Im ersten Schritt wird überprüft, ob die durch c kodierte Funktion exakt $|V_2|$ verschiedene Knoten aus V_2 und $|E_2|$ verschiedene Kanten aus E_2 enthält, also eine gültige Kodierung einer Lösung sein kann. In den beiden folgenden Schritten wird getestet, ob jede Kante und damit auch jeder Knoten aus H ein Pendant in G besitzt. Formal wird folgendes überprüft: $\{u, v\} \in E \Leftrightarrow \{f(u), f(v)\} \in E_2$. Somit wird das Zertifikat genau dann akzeptiert, wenn c eine gültige Subgraphisomorphie von H in G kodiert.

Laufzeit Der erste Schritt benötigt $\mathcal{O}(|V_2| \cdot \log(|V_2|))$ Schritte, realisierbar mittels Sortierung und nochmaligen Durchlaufs zum Auffinden von Duplikaten. Danach kommt noch ein additiver Anteil von $\mathcal{O}(|E_1|)$ hinzu. In der Summe ist die Laufzeit immer noch polynomiell in der Länge der Eingabe □

b) (150 Punkte)

Zeigen Sie, dass die Sprache SUBGRAPH $\mathcal{NP}$-vollständig ist. Kreuzen Sie zunächst an, welche der folgenden $\mathcal{NP}$-vollständigen, Ihnen bekannten Sprachen Sie für Ihre Reduktion verwenden wollen:

- □ 3SAT $:= \left\{ \langle\phi\rangle \;\middle|\; \phi \text{ ist eine erfüllbare Funktion in 3-CNF} \right\}$
- ☒ CLIQUE $:= \left\{ \langle G, k\rangle \;\middle|\; \begin{array}{l} G \text{ ist ein ungerichteter Graph mit einer Clique} \\ \text{der Größe mindestens } k \end{array} \right\}$
- □ VERTEXCOVER $:= \left\{ \langle G, k\rangle \;\middle|\; \begin{array}{l} G \text{ ist ein ungerichteter Graph mit einer} \\ k\text{-Knotenüberdeckung} \end{array} \right\}$

Führen Sie nun die Reduktion durch:

Beweis:
Wir geben im folgenden die *drei* bekannten Schritte an.

Reduktionsfunktion Die Reduktionsfunktion f sei durch folgenden Algorithmus festgelegt:

Algorithm 2 CLIQUETOSUBGRAPH

procedure CLIQUETOSUBGRAPH(G, k)
  Erzeuge $V_2 := \{1, \ldots, \frac{k(k-1)}{2}\}$ neue Knoten
  Verbinde alle Knoten untereinander und speichere dies in E_2
  return $\langle G, H = (V_2, E_2)\rangle$ ▷ H ist eine k-Clique
end procedure

Korrektheit Wir zeigen bei Richtungen:

$w \in$ CLIQUE $\Rightarrow G$ ist ein ungerichteter Graph mit einer Clique der Größe $\geq k$
$\Rightarrow$ Sei $H = (V_2, E_2)$ der von der Clique induzierte Teilgraph
$\Rightarrow$ Dann besitzt $f(w)$ genau H als Untegraph von G

$$
\begin{aligned}
f(w) \in \textsc{Subgraph} \quad & \Rightarrow G \text{ enthält einen zu } H \text{ isomorphen Subgraphen.} \\
& \quad\ \text{Dieser ist eine Clique nach Definition von } f(w) \\
& \Rightarrow w \text{ besitzt eine Clique der Größe } k \\
& \Rightarrow w \text{ besitzt eine Clique der Größe mindestens } k \\
& \Rightarrow w \in \textsc{Clique}
\end{aligned}
$$

Laufzeit Erzeugen einer k-Clique benötigt Zeit $\mathcal{O}(k^2)$, alle weiteren Operationen der Reduktion sind konstant. In der Summe ist die Reduktion also polynomiell. □

Aufgabe 5 (100 Punkte)

Beantworten Sie die folgenden Fragen ohne Begründung. Für jede richtige Antwort gibt es 10 Pluspunkte, für jede falsche 10 Minuspunkte. Nicht beantwortete Fragen erhalten null Punkte. Sie können insgesamt für diese Aufgabe nicht weniger als null Punkte erreichen.

1. Reguläre Sprachen können in linearer Zeit entschieden werden.
   □ Richtig □ Falsch

2. Jede Chomsky-1-Grammatik ist kontextfrei.
   □ Richtig □ Falsch

3. Sei L_1 eine kontextfreie Sprache. Dann gilt für $L_2 \subseteq L_1$, dass L_2 ebenfalls kontextfrei ist.
   □ Richtig □ Falsch

4. Jede Sprache in $\mathcal{NP}$ ist rekursiv aufzählbar.
   □ Richtig □ Falsch

5. Für jede Sprache, für die es eine Chomsky-0-Grammatik gibt, lässt sich auch eine nichtdeterministische Turingmaschine angeben, die diese Sprache entscheidet.
   □ Richtig □ Falsch

6. Falls es einen polynomiellen Algorithmus für SUBSET-SUM gibt, gilt $\mathcal{P} = \mathcal{NP}$.
   □ Richtig □ Falsch

7. Die Beschreibungskomplexität einer Zeichenkette ergibt sich aus der Länge der Zeichenkette multipliziert mit einem ganzzahligen Faktor c.
   □ Richtig □ Falsch

8. Gegeben sei die Sprache

   $$\text{DOZENT} := \{\langle M\rangle \mid M \text{ akzeptiert keine Eingabe außer der eigenen Codierung}\}$$

   Nach dem Satz von Rice ist die Sprache DOZENT nicht entscheidbar.
   □ Richtig □ Falsch

9. Wir wissen, dass für eine Sprache L gilt: $\overline{H} \leq_T L$. Also ist L nicht rekursiv aufzählbar.
   □ Richtig □ Falsch

10. Für alle Sprachen in $\mathcal{P}$ hat der beste bekannte Algorithmus eine Laufzeit von $O(2^n)$.
    □ Richtig □ Falsch

1. richtig
2. falsch
3. falsch
4. richtig
5. falsch
6. richtig
7. falsch
8. richtig
9. falsch
10. richtig