TU Ilmenau, Fakultät IA
Institut für Theoretische Informatik
FG Komplexitätstheorie und Effiziente Algorithmen
Univ.-Prof. Dr. Martin Dietzfelbinger, Dr. Lucia Draque Penso

Klausur Berechenbarkeit und Komplexität WS 09/10

20. Februar 2010

NICHT MIT BLEISTIFT ODER ROTSTIFT SCHREIBEN!

Heften Sie die Blätter bei Abgabe zusammen, und tragen Sie auf jedem Blatt Ihren Namen und Vornamen, Ihre Studiennummer und Matrikel ein.

ES SIND KEINE HILFSMITTEL, INSBESONDERE TASCHENRECHNER ODER MOBILTELEFONE, ZUGELASSEN!

Arbeitszeit 90min

Name, Vorname:
Studiennummer und Matrikel:

Code

abgegeben: **2 Aufgabenblätter**
. . . eigene Blätter

Einsichtnahme

Datum, Unterschrift

Aufgabe	1	2	3	4	5	6	Ges.	Zus.	Bonus	Total	Note
erreichbare Punktzahl	21	14	20	8	15(+4)	12	90	4	15	109	
erreichte Punktzahl											

Aufgabe 1 [21 Punkte]

Bitte kreuzen Sie für jede der folgenden Fragen in jeder Zeile entweder „JA“ oder „NEIN“ an.

Bewertung: Ist C die Anzahl der richtigen Antworten und F die Anzahl der falschen Antworten, so errechnet sich die Anzahl P der erzielten Punkte aus $P := \max\{0, C - F\}$. Nicht beantwortete Fragen haben auf die Punkte keinen Einfluss.

(a) Bezeichne P beliebige ASCII-Texte. Welches der folgenden Probleme ist/sind algorithmisch entscheidbar?

JA	NEIN	
[ ]	[ ]	Zu vorgegebenem P entscheide, ob P ein syntaktisch korrektes Java-Programm ist.
[ ]	[ ]	Zu vorgegebenem P und x entscheide, ob P ein syntaktisch korrektes Java-Programm ist, das auf Eingabe x die Ausgabe $Hello$ erzeugt.
[ ]	[ ]	Zu vorgegebenem P entscheide, ob P ein syntaktisch korrektes Java-Programm ist, das auf Eingabe P (als Text) hält.

(b) Um zu beweisen, dass eine Sprache L rekursiv ist, genügt es …

JA	NEIN	
[ ]	[ ]	zu zeigen, dass $L \leq L' \in \mathsf{NP}$ gilt.
[ ]	[ ]	eine nichtdeterministische Turingmaschine M mit $L = L_M$ anzugeben, so dass für alle $x \in \Sigma^*$ der Berechnungsbaum $CT_M(x)$ endlich ist.

(c) Um zu beweisen, dass eine Sprache L nicht rekursiv ist, genügt es…

JA	NEIN	
[ ]	[ ]	zu zeigen, dass $K \leq L$ gilt.
[ ]	[ ]	zu zeigen, dass $\overline{H} \leq L$ gilt.

(d) Welche der folgenden Sprachen ist/sind rekursiv?

JA	NEIN	
[ ]	[ ]	$\{\langle M \rangle \mid M$ hält auf mindestens 2010 Eingaben$\}$
[ ]	[ ]	$\{\langle M \rangle \mid M$ macht auf Eingabe $\langle M \rangle$ höchstens 2010 Schritte$\}$

(e) Um zu beweisen, dass eine Sprache L rekursiv aufzählbar ist, genügt es …

JA	NEIN	
[ ]	[ ]	zu zeigen, dass $L = L(G)$ für eine Chomsky-0-Grammatik G gilt.
[ ]	[ ]	eine nichtdeterministische Turingmaschine M mit $L = L_M$ anzugeben, so dass für alle $x \in \Sigma^*$ der Berechnungsbaum $CT_M(x)$ unendlich ist.

(f) Um zu beweisen, dass eine Sprache L nicht rekursiv aufzählbar ist, genügt es…

JA	NEIN	
[ ]	[ ]	zu zeigen, dass $\overline{K} \leq L$ gilt.
[ ]	[ ]	zu zeigen, dass $H \leq L$ gilt.

(g) Welche der folgenden Sprachen , für eine fest vorgegebene Turingmaschine M, ist/sind rekursiv aufzählbar?

JA	NEIN	
[ ]	[ ]	$\{x \in \{0,1\}^* \mid x$ hat die Form $\langle M \rangle$ und M verwirft $x\}$
[ ]	[ ]	$\{x \in \{0,1\}^* \mid x$ hat die Form $\langle M \rangle y$ und M hält nicht auf $y\}$

(h) Welche der folgenden Aussagen ist/sind korrekt (falls $\mathsf{P} \neq \mathsf{NP}$)?

JA	NEIN	
[ ]	[ ]	Wenn $L \leq_p L'$ und $L' \in \mathsf{P}$, dann gilt $L \in \mathsf{P}$.
[ ]	[ ]	Wenn $L \leq_p L_{\text{RUCKSACK}}$, dann ist L NP-schwer.

(i) Welche der folgenden Aussagen ist/sind korrekt, wenn $\mathsf{P} \neq \mathsf{NP}$ gilt?

JA	NEIN	
[ ]	[ ]	$\{L \mid L \text{ ist rekursiv}\} \not\subseteq \mathsf{P}$.
[ ]	[ ]	Das Hamiltonkreisproblem hat keinen Polynomialzeitalgorithmus.

(j) Welche der folgenden Aussagen ist/sind korrekt, wenn $\mathsf{P} = \mathsf{NP}$ gilt?

JA	NEIN	
[ ]	[ ]	$\mathsf{co} - \mathsf{NP} = \mathsf{NP}$
[ ]	[ ]	Das Binpackingproblem hat einen Polynomialzeitalgorithmus.

Aufgabe 2 (Definitionen und grundlegende Konzepte) [14 Punkte]

Geben Sie die Definitionen der folgenden Begriffe und Konzepte (wie in der Vorlesung) an beziehungsweise beantworten Sie die folgenden Fragen!

(a) [2 Punkte] Definieren Sie H und K.

(b) [2 Punkte] Wie verhält sich eine universelle Turingmaschine U auf einer Eingabe der Form $\langle M \rangle x$?

[1 Punkt] Was passiert, wenn U eine Eingabe y erhält, die nicht von der Form $\langle M \rangle x$ ist?

(c) [2 Punkte] Definieren Sie das Postsche Korrespondenzproblem (PKP) zu $\Sigma = \{0, 1\}$. (Input:..., Frage:....)

(d) [2 Punkte] Definieren Sie „$L \leq L'$ und $L \leq_p L'$“, für Sprachen L, L'.

(e) [2 Punkte] Definieren Sie „L ist NP-vollständig“!

(f) [3 Punkte] Definieren Sie „φ ist erfüllbar“ für eine KNF-Formel φ. Definieren Sie L_{SAT} und formulieren Sie den Satz von Cook.

Aufgabe 3 (Nachweis der Rekursivität) [8 Punkte]

Beweisen Sie: L und $\overline{L}$ sind rekursiv aufzählbar $\Rightarrow$ L ist rekursiv.

Aufgabe 4 (Satz von Rice: Nachweis der Nichtrekursivität) [20 Punkte]

(a) [4 Punkte] Formulieren Sie den *Satz von Rice* in der Version für Sprachen.

(b) [6 Punkte] Beweisen Sie mittels des *Satzes von Rice*, dass die nachfolgende Sprache L nicht rekursiv ist.

$$L := \{\langle M \rangle \mid M \text{ hält sowohl auf Input } 1 \text{ als auch auf Input } 0\}$$

(c) [6 Punkte] Beweisen Sie, dass L rekursiv aufzählbar ist. ***Hinweis:*** *Universelle TM U.*

(d) [4 Punkte] Beweisen Sie, dass die nachfolgende Sprache L' nicht rekursiv aufzählbar ist.

$$L' := \{\langle M \rangle \mid M \text{ hält } \underline{\text{nicht}} \text{ sowohl auf Input } 1 \text{ als auch auf Input } 0\}$$

Hinweis: *Benutzen Sie (b), (c) und die Aussage von Aufgabe 3.*

Aufgabe 5 ($L_{3-\mathrm{SAT}}$ ist NP-Vollständig) [15(+4) Punkte]

(a) [6 Punkte] Zeigen Sie: $L_{3-\mathrm{SAT}} \in \mathsf{NP}$.

(b) [5 Punkte] In der Vorlesung wurde für die Reduktion $L_{\mathrm{SAT}} \leq_p L_{3-\mathrm{SAT}}$ eine Reduktionsfunktion f angegeben. Wie wird die folgende KNF-Formel φ von f transformiert?

$$\text{Eingabe: } \varphi = (X_1) \wedge (X_2 \vee X_3 \vee \overline{X_4} \vee X_5) \wedge (X_1 \vee \overline{X_2}) \wedge (X_2 \vee X_3 \vee \overline{X_4} \vee X_5 \vee X_6)$$

$$\text{Ausgabe: } f(\varphi) = \varphi^* = \cdots ?$$

(c) [4 Punkte] Formulieren Sie die beiden Aussagen, die man über f beweisen muss, um „$L_{\mathrm{SAT}} \leq_p L_{3-\mathrm{SAT}}$ via f" zu erhalten.

(d) [4 Zusatzpunkte] Gegeben sei die für φ erfüllende Belegung $v(X_1) = v(X_2) = v(X_3) = v(X_4) = 1$ und $v(X_5) = v(X_6) = 0$. Wie sieht die entsprechende erfüllende Belegung v^* für φ^* aus (b) aus?

Aufgabe 6 (Primitive Rekursion, $\mathcal{PR} \subseteq \mathcal{R}$ aber $\mathcal{R} \not\subseteq \mathcal{PR}$) [12 Punkte]

Die Klasse $\mathcal{PR}$ erhält man durch iterierte Anwendung der Regeln *Primitive Rekursion* und *Einsetzen* auf die Basisfunktionen *Konstante*, *Nachfolger* und *Projektion*.

Wir wissen: Jede Funktion in $\mathcal{PR}$ kann durch einen endlichen Text über einem endlichen Alphabet beschrieben werden. Dabei ist die Menge dieser Texte entscheidbar. (Man kann algorithmisch entscheiden, ob ein Text eine Funktionenbeschreibung - ein Programm - ist oder nicht.)

Zeigen Sie:

(a) [6 Punkte] Man kann die Funktionen in $\mathcal{PR}$ so als $f_0, f_1, f_2, \ldots$ durchnummerieren, dass aus bin(i) die Beschreibung von f_i berechnet werden kann.

(b) [6 Punkte] Es gibt eine berechenbare Funktion g, die nicht in $\mathcal{PR}$ ist. (Das bedeutet: $\mathcal{R} \not\subseteq \mathcal{PR}$.) ***Hinweis:*** *Definieren Sie die Funktion $g : \mathbb{N} \to \mathbb{N}$ durch Diagonalisierung. Beschreiben Sie, wie man aus i den Wert $g(i)$ berechnen kann.*

Viel Erfolg!