TU Ilmenau, Fakultät IA
Institut TTI, FG Komplexitätstheorie und Effiziente Algorithmen
PD Dr. rer. nat. habil. K.-H. Niggl

Klausur „Komplexitätstheorie“ WS03

19. Februar 2004

NICHT MIT BLEISTIFT ODER ROTSTIFT SCHREIBEN!

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

Name, Vorname:
Studiennummer und Matrikel:

Code

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

Einsichtnahme

Datum, Unterschrift

Aufgabe	1	2	3	4	5	6	7	Ges.
erreichbare Punktzahl	30	12	16	8	12	12	10	100
erreichte Punktzahl								

Es sind keine Hilfsmittel, insbesondere Taschenrechner oder Mobiltelefone, zugelassen.

Aufgabe 1 (Fragenkatalog) [30 Punkte]

Bitte kreuzen Sie für jede der folgenden 30 Fragen entweder „JA“ oder „NEIN“ an.
Bewertung: Ist C die Anzahl der richtigen Antworten, so errechnet sich die Anzahl P der erzielten Punkte aus $P := \frac{3}{2} \cdot \max\{0, C - 10\}$. Insbesondere: Kein Kreuz wird als falsche Antwort gewertet.

(a) Welche der folgenden Aussagen ist/sind korrekt?

JA NEIN
[ ] [ ] Wenn $P \leq_N P'$ (mit O-Konstante c) und $T_P(n) \geq d \cdot n \cdot \log n - O(n)$ für genügend große n, dann gilt $T_{P'}(n) \geq \frac{d}{c} \cdot n \cdot \log n - O(n)$ für genügend große n.
[ ] [ ] Wenn $P \leq_N P'$ und P ist effizient lösbar, dann ist P' effizient lösbar.
[ ] [ ] Wenn $P \leq_N P'$ und $T_P(n) = \Omega(n \cdot \log n)$, dann gilt $T_{P'}(n) = \Omega(n \cdot \log n)$.

(b) Welche der nachstehenden Aussagen folgt/folgen aus dem Satz von Ben-Or?

JA NEIN
[ ] [ ] Jedes RRAM-Programm für ELEMENT-UNIQUENESS benötigt für genügend große n mindestens $0.38 \cdot n \cdot \log n - O(n)$ Schritte.
[ ] [ ] Für jedes RRAM-Programm M für das Sortieren von reellen Zahlen gilt $T_M(n) = \Omega(n^2)$.
[ ] [ ] Für jedes RRAM-Programm M zur Berechnung der konvexen Hülle von Punkten in der Ebene gilt $T_M(n) = \Omega(n \cdot \log n)$.

(c) Welche der folgenden Aussagen ist/sind korrekt?

JA NEIN
[ ] [ ] Wenn $A \leq_T B$, so gibt es Polynome p, q, r mit $T_A(n) \leq p(n) + q(n) \cdot T_B(r(n))$.
[ ] [ ] Wenn $A \leq_T B$ und B besitzt einen Polynomialzeitalgorithmus, so auch A.
[ ] [ ] Wenn $\mathsf{P} \neq \mathsf{NP}$, dann besitzt $\text{CLIQUE}_{\text{strukopt}}$ keinen Polynomialzeitalgorithmus.

(d) Welche der folgenden Aussagen ist/sind korrekt?

JA NEIN

[ ] [ ] Wenn $L \leq_p L'$ und $L' \in \mathsf{NP}$, dann gilt $L \in \mathsf{NP}$.

[ ] [ ] Wenn $L \neq L_M$ für jede $O(2^{n^k})$-zeitbeschränkte TM M, dann gilt $L \notin \mathsf{NP}$.

[ ] [ ] Wenn $L \leq_p$ SAT, dann ist L NP-vollständig.

(e) Welche der folgenden Aussagen ist/sind korrekt?

JA NEIN

[ ] [ ] Wenn $\mathsf{P} \neq \mathsf{NP}$, dann besitzt keine NPC-Sprache einen Polynomialzeitalgorithmus.

[ ] [ ] Wenn $\mathsf{P} = \mathsf{NP}$, dann gibt es eine Sprache $L \in \mathsf{P}$ mit $L' \leq_p L$ für alle $L' \in \mathsf{P}$.

[ ] [ ] Wenn $\mathsf{P} \neq \mathsf{NP}$, dann gilt nicht 3-SAT $\leq_p$ RUCKSACK*.

(f) Welche der folgenden Aussagen ist/sind für **alle** Sprachen $L \subseteq \Sigma^*$ mit $\# \notin \Sigma$ korrekt?

JA NEIN

[ ] [ ] Wenn $L \in \mathsf{NP}$, dann gilt $L = \{x \in \Sigma^* \mid \exists y \in \{0,1\}^{\leq p(|x|)} \colon x\#y \in L_0\}$ für ein Polynom p und eine Sprache $L_0 \in \mathsf{P}$.

[ ] [ ] Wenn $L \in \mathsf{P}$, dann gilt $L \leq_p \{1\}$.

[ ] [ ] Wenn $L = \{x \in \Sigma^* \mid \forall y \in \{0,1\}^{\leq p(|x|)} \colon x\#y \in L_0\}$ für ein Polynom p und $L_0 \in \mathsf{P}$, dann gilt $\overline{L} \in \mathsf{NP}$.

(g) Welche der folgenden Aussagen ist/sind korrekt?

JA NEIN

[ ] [ ] Für den Algorithmus GRS (Greedy Rucksack) gilt $\mathcal{R}^{\infty}_{\mathsf{GRS}} < 3$.

[ ] [ ] Für den Algorithmus MGRS (Maximum Greedy Rucksack) gilt $\mathcal{R}_{\mathsf{MGRS}} \leq 2$.

[ ] [ ] Für den Algorithmus FFD (First Fit Decreasing) gilt stets: $r_{\mathsf{FFD}}(x) < \frac{3}{2} + \frac{1}{m^*(x)}$

(h) Welche der folgenden Aussagen ist/sind korrekt?

JA NEIN

[ ] [ ] Es ist ein offenes Problem, ob MIN-BINPACKING einen Approximationsalgorithmus mit worst-case Güte 1.4 besitzt.

[ ] [ ] Es ist ein offenes Problem, ob MIN-TSP einen Approximationsalgorithmus besitzt.

[ ] [ ] Es ist ein offenes Problem, ob MIN-PARTITION ein polynomielles Approximationsschema besitzt.

(i) Welche der folgenden Aussagen ist/sind für **alle** NPO-Probleme $\mathcal{P}, \mathcal{P}'$ korrekt?

JA NEIN

[ ] [ ] Wenn $\mathcal{P}_D \in \mathsf{NPC}$, dann gilt $L \leq_T \mathcal{P}_C$ für alle $L \in \mathsf{NP}$.

[ ] [ ] Wenn $\mathcal{P}_D$ keinen Polynomialzeitalgorithmus besitzt, dann auch nicht $\mathcal{P}_C$.

[ ] [ ] Wenn $\neg(\mathcal{P}_C \leq_T \mathcal{P}_D)$, dann gilt $\mathcal{P}_D \notin \mathsf{NPC}$.

(j) Welche der folgenden Aussagen ist/sind korrekt?

JA NEIN

[ ] [ ] Für alle Sprachen L gilt: $\overline{L} \in \mathsf{co-NPC} \Leftrightarrow L \in \mathsf{NPC}$

[ ] [ ] Wenn $\mathsf{NP} \neq \mathsf{co-NP}$, dann gilt $\mathsf{P} \neq \mathsf{NP}$ und $\mathsf{NPC} \cap \mathsf{co-NP} \neq \emptyset$.

[ ] [ ] Wenn MIN-DNF einen Polynomialzeitalgorithmus besitzt, so auch DNF-TAUTOLOGIE und es gilt $\mathsf{P} = \mathsf{NP}$.

<u>Aufgabe 2</u> (Definitionen von Begriffen und Konzepten aus der Vorlesung) [12 Punkte]

(a) Definieren Sie den Begriff $L \leq_p L'$ (*Sprache L ist polynomialzeitreduzierbar auf Sprache L'*).

(b) Definieren Sie den Begriff L *ist NP-vollständig.*

(c) Definieren Sie den Begriff $A \leq_T B$ (*Problem A ist turingreduzierbar auf Problem B*).

(d) Definieren Sie den Begriff $\mathcal{A}$ *ist Approximationsalgorithmus für ein NPO-Problem* $\mathcal{P}$.

(e) Definieren Sie die *Güte* $r_{\mathcal{A}}$ eines Approximationsalgorithmus $\mathcal{A}$ für ein NPO-Problem $\mathcal{P}$.

(f) Definieren Sie die *Sprachklasse* $\mathsf{co}-\mathsf{NP}$.

<u>Aufgabe 3</u> (Sätze bzw. Konstruktionen aus der Vorlesung) [8+8 Punkte]

(a) Sei $\varphi \equiv C_1 \wedge \ldots \wedge C_r$ $(r \geq 1)$ mit $C_i \equiv (l_{i1} \vee \ldots \vee l_{is_i})$, $s_i \geq 4$ (für $i = 1, \ldots, r$) eine beliebige KNF-Formel. Definieren Sie eine 3-KNF-Formel $\widehat{\varphi}$ mit

$$\varphi \text{ ist erfüllbar} \Leftrightarrow \widehat{\varphi} \text{ ist erfüllbar}$$

und beweisen Sie davon die Implikation „$\Rightarrow$“.

(b) Sei $\varphi \equiv C_1 \wedge \ldots \wedge C_r$ $(r \geq 1)$ mit $C_i \equiv (l_{i1} \vee l_{i2} \vee l_{i3})$ (für $i = 1, \ldots, r$) eine beliebige 3-KNF-Formel. Definieren Sie einen Graphen G_φ mit

$$\varphi \text{ ist erfüllbar} \Leftrightarrow G_\varphi \text{ besitzt eine Clique der Größe } r$$

und beweisen Sie davon die Implikation „$\Rightarrow$“.

<u>Aufgabe 4</u> (Sätze bzw. Konstruktionen aus der Vorlesung) [8 Punkte]

MIN-TSP ist das folgende NPO-Problem: Zu gegebener Matrix $M := (a_{ij}) \in \mathbb{N}^{m \times m}$ $(m \geq 1)$ von Entfernungen a_{ij} zwischen Stadt i und Stadt j, berechne eine *Rundreise durch die m Städte*, d.h. eine Permutation π von $\{1, \ldots, m\}$, mit minimalen *Kosten* $m(M, \pi) := (\sum_{1 \leq i < n} a_{\pi(i)\pi(i+1)}) + a_{\pi(m)\pi(1)}$.

Sei $\mathcal{A}$ ein Approximationsalgorithmus für MIN-TSP mit:

$(*)$ Für jede Matrix $M := (a_{ij}) \in \mathbb{N}^{m \times m}$ $(m \geq 1)$ gilt: $m_{\mathcal{A}}(M) \leq 2^m \cdot m^*(M)$

wobei $m_{\mathcal{A}}(M)$ die Kosten der von $\mathcal{A}$ auf Eingabe M berechneten Rundreise bezeichnet und $m^*(M)$ die Kosten einer optimalen Rundreise.

Konstruieren Sie aus $\mathcal{A}$ einen Polynomialzeitalgorithmus $\mathcal{B}$, so daß für alle Graphen $G = (V, E)$ gilt

$(**)$ G besitzt einen Hamiltonkreis $\Leftrightarrow$ $\mathcal{B}$ akzeptiert Eingabe G

und beweisen Sie die Aussage $(**)$.

Aufgabe 5 (Reduktionsmethode für NPC-Sprachen) [12 Punkte]

CONFLICT ist, informal beschrieben, das folgende Problem: Gegeben $m \geq 1$ Objekte mit *Nutzenwerten* $c_1, \ldots, c_m \in \mathbb{N}$ sowie eine Menge von *Konfliktpaaren* $K \subseteq \{1, \ldots, m\} \times \{1, \ldots, m\}$ und eine *Nutzenschranke* $c \in \mathbb{N}$, entscheide, ob eine *konfliktfreie Auswahl* $I \subseteq \{1, \ldots, m\}$, d.h.

$$\forall i, j \in I : i \neq j \Rightarrow (i, j) \notin K,$$

mit *Gesamtnutzen* $m(I) := \sum_{i \in I} c_i \geq c$ existiert. Formal erhalten wir die Sprache

$$\text{CONFLICT} := \{\text{bin}(c_1)\# \ldots \text{bin}(c_m)\#\langle K \rangle\#\text{bin}(c) \mid \exists \text{ konfliktfreie Auswahl } I \text{ mit } m(I) \geq c\}$$

wobei $\langle K \rangle := a_{11} \ldots a_{1m} \ldots a_{m1} \ldots a_{mm}$ die Adjazenzmatrix von K ist.

Zeigen Sie: CONFLICT ist NP-vollständig.

Aufgabe 6 (Approximationsalgorithmen) [12 Punkte]

Betrachten Sie den folgenden Approximationsalgorithmus ANY FIT (AF) für das BINPACKING-Problem.

> **Eingabe:** $(a_1, \ldots, a_n, b) \in \mathbb{Q}^{n+1}$ mit $n \geq 1$ und $0 < a_i \leq b$ für $i = 1, \ldots, n$
> **Ausgabe:** Aufteilung $r_1, \ldots, r_n \in \{1, \ldots, n\}$ der n Objekte in Bins mit Kapazität b
> **for** $i := 1, 2, \ldots, n$ **do**
> $r_i :=$ Nummer irgendeines angefangenen Bins, in das „Objekt“ a_i noch paßt, falls ein solches existiert, andernfalls $r_i :=$ kleinste Nummer eines leeren Bins.

Sei $m_{\mathsf{AF}}(x)$ die Anzahl der Behälter, die der Algorithmus AF auf Eingabe x benutzt, und $m^*(x)$ die Anzahl der Behälter einer optimalen Verteilung für x.

Zeigen Sie: Für alle Eingaben x gilt: $\frac{m_{\mathsf{AF}}(x)}{m^*(x)} < 2$

Aufgabe 7 (Turingreduzierbarkeit) [10 Punkte]

Eine Matrix $A := (a_{ij}) \in \{0, 1\}^{m \times n}$ heiße *zulässig*, falls jede Spalte von A eine Eins enthält, d.h.

$$\forall j \in \{1, \ldots, n\} \, \exists i \in \{1, \ldots, m\} : a_{ij} = 1.$$

Für $I \subseteq \{1, \ldots, m\}$ bezeichne $A(I)$ die Matrix, die aus A entsteht, indem alle Zeilen $(a_{k1}, \ldots, a_{kn})$ mit $k \in \{1, \ldots, m\} \setminus I$ gestrichen werden.

- SET-COVER$_{\text{strukopt}}$ ist das folgende Problem: Zu gegebener zulässiger Matrix A berechne eine (bzgl. der Größe) minimale Auswahl $I \subseteq \{1, \ldots, m\}$ von Zeilen, so daß die Matrix $A(I)$ zulässig ist.
- SET-COVER$_{\text{paropt}}$ ist das folgende Problem: Zu gegebener zulässiger Matrix A berechne die Größe $m^*(A)$ einer minimalen Auswahl $I \subseteq \{1, \ldots, m\}$, für die $A(I)$ zulässig ist.

Zeigen Sie: SET-COVER$_{\text{strukopt}}$ $\leq_T$ SET-COVER$_{\text{paropt}}$

Viel Erfolg!