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

Klausur „Komplexitätstheorie“ WS07/08

27. Februar 2008

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 nachstehenden Aussagen folgt/folgen aus dem Satz von Ben-Or?

JA NEIN
[ ] [ ] Für jedes RRAM-Programm M für ELEMENT UNIQUENESS (kurz EU) gilt $T_M(n) \geq 0.38 \cdot \log(\#\mathrm{EU}_n) - 0.68 \cdot n$ für genügend große n.
[ ] [ ] Jedes RRAM-Programm für das Sortieren von n reellen Zahlen macht für genügend große n im schlechtesten Fall mindestens $0.38 \cdot n \cdot \log n - O(n)$ Schritte.
[ ] [ ] Für jedes RRAM-Programm M zur Berechnung der konvexen Hülle von n Punkten in der Ebene gilt $T_M(n) \geq 0.16 \cdot n \cdot \log n - O(n)$ für genügend große n.

(b) Welche der folgenden Aussagen ist/sind korrekt, wenn $P \leq_N P'$ mit O-Konstante c gilt?

JA NEIN
[ ] [ ] Aus jedem RRAM-Programm M' für P' kann man ein RRAM-Programm M für P mit $T_{M'}(c \cdot n) \geq T_M(n) - O(n \cdot \log n)$ konstruieren.
[ ] [ ] Wenn P einen Polynomialzeitalgorithmus besitzt, dann auch P'.
[ ] [ ] Aus $T_P(n) \geq d \cdot n \cdot \log n - O(n)$ folgt $T_{P'}(n) \geq \frac{d}{c} \cdot n \cdot \log n - O(n)$.

(c) Welche der folgenden Aussagen ist/sind korrekt, wenn $A \leq_T B$ gilt?

JA NEIN
[ ] [ ] Es existieren Polynome q, r derart, daß man aus jeder Turingmaschine N für B eine Turingmaschine M für A mit $T_M(n) \leq q(n) \cdot (1 + T_N(r(n)))$ konstruieren kann.
[ ] [ ] Wenn $B \leq_p C$ gilt, dann gilt $A \leq_T C$.
[ ] [ ] Wenn $B \leq_T A$ gilt, dann besitzt A einen Polynomialzeitalgorithmus genau dann, wenn dies für B gilt.

(d) Welche der folgenden Aussagen ist/sind für **alle** Sprachen L korrekt?

JA NEIN

[ ] [ ] Wenn $L \leq_p$ CLIQUE gilt, dann ist L NP-schwer.

[ ] [ ] Wenn L von keiner $O(2^{n^k})$-zeitbeschränkten TM akzeptiert wird, so gilt $L \notin \mathsf{NP}$.

[ ] [ ] Wenn SAT $\leq_p L$ und $L \leq_p$ DHC gilt, dann gilt $L \in \mathsf{NPC}$.

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

JA NEIN

[ ] [ ] Wenn $\mathsf{P} = \mathsf{NP}$ gilt, dann gilt 3-COLORING $\leq_p \{0\}$.

[ ] [ ] Wenn RUCKSACK $\in \mathsf{P}$ gilt, dann gilt $\mathsf{P} = \mathsf{NP}$.

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

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

JA NEIN

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

[ ] [ ] Aus $\overline{L} = \{x \in \Sigma^* \mid \forall y \in \{0,1\}^{\leq |x|^2} \colon x\#y \in L_0\}$ und $L_0 \in \mathsf{P}$ folgt $L \in \mathsf{NP}$.

[ ] [ ] Aus $\overline{L} = \{x \in \Sigma^* \mid \exists y \in \{0,1\}^{\leq |x|^2} \colon x\#y \in L_0\}$ und $L_0 \in \mathsf{P}$ folgt $L \in$ co-NP.

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

JA NEIN

[ ] [ ] MAX-RUCKSACK ist 2-approximierbar via Algorithmus GRS (Greedy Rucksack).

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

[ ] [ ] MIN-BINPACKING ist asymptotisch $\frac{3}{2}$-approximierbar via Algorithmus FFD (First Fit Decreasing).

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

JA NEIN

[ ] [ ] MIN-GRAPH-COLORING besitzt keinen Approximationsalgorithmus $\mathcal{A}$ mit $\mathcal{R}^\infty_{\mathcal{A}} < \frac{4}{3}$, es sei denn $\mathsf{P} = \mathsf{NP}$.

[ ] [ ] MIN-TSP besitzt keinen Approximationsalgorithmus, es sei denn NP = co-NP.

[ ] [ ] MIN-PARTITION besitzt kein polynomielles Approximationsschema, es sei denn $\mathsf{P} = \mathsf{NP}$.

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

JA NEIN

[ ] [ ] Wenn $\mathcal{P}_D$ NP-schwer ist, dann gilt $\mathcal{P}_D \in \mathsf{NPC}$.

[ ] [ ] Wenn nicht $\mathcal{P}_C \leq_T \mathcal{P}_D$ gilt, dann gilt auch nicht $\mathcal{P}_D \leq_T \mathcal{P}_E$.

[ ] [ ] Wenn $\mathcal{P}_D$ NP-schwer ist, dann gilt $\mathcal{P}_D =_T \mathcal{P}_E =_T \mathcal{P}_C$.

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

JA NEIN

[ ] [ ] Es gilt $L \in$ co-NPC genau dann, wenn CLIQUE $\leq_p \overline{L}$ und $\overline{L} \leq_p$ PARTITION gilt.

[ ] [ ] Es gilt NP = co-NP genau dann, wenn $\overline{\text{SAT}} \in \mathsf{NP}$ gilt.

[ ] [ ] Aus $L \in \Sigma^p_k \cap \Pi^p_{k+1}$ folgt $L \in \Sigma^p_{k+2} \cap \Pi^p_{k+2}$.

<u>Aufgabe 2</u> (Definitionen von Begriffen aus der Vorlesung) [$12 = 2 + 2 + 2.5 + 1.5 + 2 + 2$ Punkte]

Im folgenden sei $\mathcal{P} = (D, O, SOL, m, goal)$ ein beliebiges, aber fest gewähltes NPO-Problem.

(a) Definieren Sie den Begriff $L \leq_p L'$, für $L \subseteq \Sigma^*$ und $L' \subseteq \Delta^*$.

(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* $\mathcal{P}$.

(e) Definieren Sie die *Güte* $r_{\mathcal{A}}(x)$ eines Approximationsalgorithmus $\mathcal{A}$ für $\mathcal{P}$ bzgl. Input $x \in D$.

(f) Definieren Sie die *Entscheidungsvariante* $\mathcal{P}_D$ für $\mathcal{P}$.

<u>Aufgabe 3</u> (Sätze bzw. Konstruktionen aus der Vorlesung) [$16 = 9 + 7$ Punkte]

(a) Sei $C \equiv (l_1 \vee \ldots \vee l_s)$ eine beliebige Klausel (von Literalen $l_1, \ldots, l_s$) mit $s \geq 4$.
Definieren Sie eine 3-KNF-Formel $\widehat{C}$ mit der Eigenschaft

$$C \text{ ist erfüllbar} \iff \widehat{C} \text{ ist erfüllbar}$$

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

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

$$\varphi \text{ ist erfüllbar} \iff G_\varphi \text{ besitzt eine } r\text{-Clique}$$

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

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

Das NPO-Problem MIN-TSP besteht, informal beschrieben, in der Aufgabe, zu gegebenen Entfernungen $(c_{ij})_{1 \leq i,j \leq n}$ zwischen n Städten eine *billigste Rundreise* zu berechnen.

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

$$(*) \qquad \text{Für jeden Input } x = (c_{ij})_{1 \leq i,j \leq n} \text{ gilt:} \quad \frac{m_{\mathcal{A}}(x)}{m^*(x)} \leq 2^n$$

Dabei bezeichne $m_{\mathcal{A}}(x)$ den Wert (Gesamtstrecke) der von $\mathcal{A}$ auf Input x berechneten Rundreise $\mathcal{A}(x) \in S_n$ und $m^*(x)$ den Wert einer minimalen Rundreise bzgl. Input x.

Konstruieren Sie mittels $\mathcal{A}$ einen *Polynomialzeitalgorithmus* $\mathcal{B}$ für HAMILTONIAN CIRCUIT und beweisen Sie die *Korrektheit von* $\mathcal{B}$:

$$G \text{ besitzt einen Hamiltonkreis} \iff \mathcal{B} \text{ akzeptiert die Eingabe } G$$

wobei $G = (V, E)$ mit $V = \{1, \ldots, m\}$ ein beliebiger, aber fest vorgegebener Graph ist.

Aufgabe 5 (Reduktionsmethode für NPC-Sprachen) [12 = 8 + 4 Punkte]

Das Problem BASE besteht (informal beschrieben) in folgender Aufgabe: Gegeben ein Graph $G = (V, E)$ und m *Gewichte* $c_1, \ldots, c_m$ für die Knoten von G sowie eine *Gewichtsschranke* c, entscheide, ob eine *Basis* I *für* G existiert, d.h. eine Auswahl $I \subseteq V = \{1, \ldots, m\}$ mit der Eigenschaft

$$\forall (i,j) \in E\colon i \in I \text{ oder } j \in I,$$

deren *Gesamtgewicht* $m(I) := \sum\limits_{i \in I} c_i$ höchstens c beträgt. Formal erhalten wir die Sprache

$$\textsc{base} := \{\langle G\rangle\#\mathrm{bin}(c_1)\#\ldots\mathrm{bin}(c_m)\#\mathrm{bin}(c) \mid \exists \text{ Basis } I \text{ für } G \text{ mit } m(I) \leq c\}.$$

Zeigen Sie: BASE ist NP-vollständig.

Aufgabe 6 (Reduktionsmethode für NPC-Sprachen) [12 Punkte]

Betrachten Sie die 3-KNF-Formel

$$\varphi \equiv (x_1 \vee \overline{x_2} \vee \overline{x_2}) \wedge (\overline{x_1} \vee x_2 \vee x_3) \wedge (x_3 \vee x_3 \vee x_4)$$

und die φ erfüllende Belegung v mit $v(x_1) = v(x_3) = 1$ und $v(x_2) = v(x_4) = 0$.

(a) Geben Sie den gemäß Vorlesung zu φ konstruierten gerichteten Graphen G_φ mit

$$\varphi \text{ ist erfüllbar } \iff G_\varphi \text{ besitzt einen Hamiltonkreis}$$

in graphischer Darstellung an, mit Kennzeichnung der relevanten Komponenten, wie zum Beispiel die Bauteile K_i sowie die x_j- und $\overline{x_j}$-Wege.

(b) Kennzeichnen Sie in G_φ (mit geeigneter Farbe) den gemäß Vorlesung bestimmten Hamiltonkreis, der zur erfüllenden Belegung v von φ gehört.

Aufgabe 7 (Approximationsalgorithmen) [10 Punkte]

Betrachten Sie den folgenden Approximationsalgorithmus BF (Best Fit) für MIN-BINPACKING.

> **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:** Verteilung $r_1, \ldots, r_n \in \{1, \ldots, n\}$ der n Objekte in Bins mit Kapazität b
> **for** $i := 1, 2, \ldots, n$ **do**
> $r_i :=$ Nummer eines *maximal* beladenen Bins, in das „Objekt“ a_i noch paßt, falls ein solches Bin existiert, andernfalls $r_i :=$ kleinste Nummer eines leeren Bins.

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

Zeigen Sie: $\frac{m_{\mathsf{BF}}(x)}{m^*(x)} < 2$ für alle Eingaben x.

Viel Erfolg!