



## Wiederholungsklausur „Komplexitätstheorie“ SS06

1. August 2006

**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
<hr style="border: none; border-top: 1px solid black;"/>
Datum, Unterschrift

Aufgabe	1	2	3	4	5	6	7	Ges.
erreichbare Punktzahl	30	11	11	10	10	17	11	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

- Die worst case Laufzeit jedes RRAM-Programms für ELEMENT-UNIQUENESS (kurz EU) beträgt  $\Omega(\log n!)$  für  $n$  reelle Zahlen aus EU.
- Die worst case Laufzeit jedes RRAM-Programms für das Sortieren von  $n$  reellen Zahlen ist  $\Omega(n^2 \cdot \log n)$ .
- Die worst case Laufzeit jedes RRAM-Programms zur Berechnung der konvexen Hülle von  $n$  Punkten in der Ebene beträgt  $\Omega(\frac{0.38}{2} \cdot n \cdot \log n - O(n))$ .

(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  $T_M(n) - O(n) \leq T_{M'}(c \cdot n)$  konstruieren.
- Wenn  $P$  effizient lösbar ist, dann ist auch  $P'$  effizient lösbar.
- Wenn  $T_P(n) \geq d \cdot n \log n - O(n)$  für eine Konstante  $d > 0$  gilt, dann gilt  $T_{P'}(n) < \frac{d}{c} \cdot n \log n - O(n)$ .

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

JA NEIN

- Es existieren Polynome  $p, q, r$ , so daß man aus jeder Turingmaschine  $N$  für  $B$  eine Turingmaschine  $M$  für  $A$  mit  $T_M(n) \leq p(n) + q(n) \cdot T_N(r(n))$  konstruieren kann.
- Wenn  $A \leq_p B$  gilt, so folgt  $A = B$ .
- Wenn  $B$  einen Polynomialzeitalgorithmus besitzt, dann auch  $A$ .

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

JA NEIN

- Wenn  $L \leq_p L'$  und  $L \in \text{NP}$  gilt, dann gilt  $L' \in \text{NP}$ .  
  Wenn  $L$  von keiner  $O(2^{n^k})$ -zeitbeschränkten TM akzeptiert wird, so gilt  $L \notin \text{NP}$ .  
  Wenn  $\text{SAT} \leq_p L$  gilt, dann ist  $\bar{L}$  NP-schwer.

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

JA NEIN

- Wenn  $\text{P} \neq \text{NP}$  gilt, dann gilt nicht  $3\text{-SAT} \leq_p 3\text{-COLORING}$ .  
  Wenn  $\text{P} = \text{NP}$  gilt, dann gilt  $3\text{-SAT} \leq_T L$  für jede nichttriviale Sprache  $L$ .  
  Wenn  $\text{CLIQUE} \in \text{P}$  gilt, dann folgt  $\text{P} = \text{NP}$ .

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

JA NEIN

- Wenn  $L \in \text{P}$  gilt, dann gilt  $L \leq_p \{\#\}$ .  
  Es gilt  $L \in \text{NP}$  genau dann, wenn ein Polynom  $p(n)$  und eine Sprache  $L_0 \in \text{P}$  existieren, so daß für alle  $x \in \Sigma^*$  gilt:  $x \in L \iff \exists y \in \{0, 1\}^{\leq p(|x|)} : x\#y \in L_0$   
  Wenn  $L = \{x \in \Sigma^* \mid \forall y \in \{0, 1\}^* : |y| \leq p(|x|) \Rightarrow x\#y \in L_0\}$  gilt, für ein Polynom  $p(n)$  und  $L_0 \in \text{P}$ , dann gilt  $\bar{L} \in \text{NP}$ .

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

JA NEIN

- Für den Algorithmus GRS (Greedy Rucksack) gilt  $\mathcal{R}_{\text{GRS}}^\infty \leq 2$ .  
  Für den Algorithmus MGRS (Maximum Greedy Rucksack) gilt  $\mathcal{R}_{\text{MGRS}} = \infty$ .  
  Für den Algorithmus FF (First Fit) gilt  $\mathcal{R}_{\text{FF}}^\infty = 1$ .

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

JA NEIN

- MIN-PARTITION besitzt kein polynomielles Approximationsschema, es sei denn  $\text{P} = \text{NP}$ .  
  MIN-BINPACKING besitzt keinen Approximationsalgorithmus mit asymptotischer Güte  $< \frac{3}{2}$ , es sei denn  $\text{P} = \text{NP}$ .  
  MIN-TSP besitzt keinen Approximationsalgorithmus, es sei denn  $\text{P} = \text{NP}$ .

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

JA NEIN

- Wenn nicht  $\mathcal{P}_C \leq_T \mathcal{P}_D$  gilt, dann ist  $\mathcal{P}_D$  nicht NP-schwer.  
  Wenn  $\mathcal{P}_D, \mathcal{P}_C, \mathcal{P}_E$  nicht berechnungsäquivalent sind, dann ist  $\mathcal{P}_D$  nicht NP-schwer.  
  Wenn  $\mathcal{P}_D$  NP-schwer ist, dann gilt  $\mathcal{P}_D \in \text{NPC}$ .

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

JA NEIN

- $\text{NP} = \text{co-NP} \iff \text{NP}$  ist abgeschlossen unter Komplementbildung.  
   $L \notin \Sigma_{k+1}^p \cap \Pi_{k+1}^p \implies L \notin \Sigma_k^p \cup \Pi_k^p$ .  
   $\text{NP} \neq \text{co-NP} \implies \text{P} \neq \text{NP}$

**Aufgabe 2** (Definitionen von Begriffen und Konzepten aus der Vorlesung) [11 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*  $\text{co-NP}$ .

**Aufgabe 3** (Sätze bzw. Konstruktionen aus der Vorlesung) [11 Punkte]

Sei  $\varphi \equiv C_1 \wedge \dots \wedge C_r$  eine beliebige KNF-Formel, in der jede Klausel  $C_i \equiv (l_{i1} \vee \dots \vee l_{is_i})$  aus *mindestens* 4 Literalen besteht ( $s_i \geq 4$ ). Definieren Sie eine 3-KNF-Formel  $\hat{\varphi}$ , so daß gilt (mit Beweis!):

$$\varphi \text{ ist erfüllbar} \iff \hat{\varphi} \text{ ist erfüllbar}$$

**Aufgabe 4** (Sätze bzw. Konstruktionen aus der Vorlesung) [10 Punkte]

Das NPO-Problem MIN-TSP besteht, informal beschrieben, in folgender Aufgabe: Zu gegebener Matrix  $x = (c_{ij})_{1 \leq i, j \leq n}$  von Entfernungen  $c_{ij} \in \mathbb{N}$  zwischen  $n$  Städten berechne eine *billigste Rundreise*, d.h. eine Permutation  $\pi$  von  $\{1, \dots, n\}$  mit minimalen *Kosten*  $m(x, \pi) := (\sum_{1 \leq i < n} c_{\pi(i)\pi(i+1)}) + c_{\pi(n)\pi(1)}$ .

Sei  $\mathcal{A}$  ein Approximationsalgorithmus für MIN-TSP, so daß für jeden Input  $x = (c_{ij}) \in \mathbb{N}^{n \times n}$  gilt:

$$(*) \quad r_{\mathcal{A}}(x) = \frac{m_{\mathcal{A}}(x)}{m^*(x)} \leq 2^n$$

(die Güte von  $\mathcal{A}$  darf exponentiell in der Anzahl der Städte wachsen). Dabei bezeichne  $m_{\mathcal{A}}(x)$  die Kosten der von  $\mathcal{A}$  auf Eingabe  $x$  berechneten Rundreise und  $m^*(x)$  die Kosten einer optimalen Rundreise.

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

$$(**) \quad G \text{ besitzt einen Hamiltonkreis} \iff \mathcal{B} \text{ akzeptiert Eingabe } G$$

und beweisen Sie (\*\*).

**Aufgabe 5** (Approximationsalgorithmen) [10 Punkte]

Betrachten Sie den Approximationsalgorithmus WORST FIT (WF) für das BINPACKING-Problem.

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

Sei  $m_{\text{WF}}(x)$  die Anzahl der Behälter, die der Algorithmus WF auf Eingabe  $x$  (in obigem Sinne) benutzt, und  $m^*(x)$  die Anzahl der Behälter einer optimalen Verteilung für  $x$ .

Zeigen Sie: Für alle Eingaben  $x$  (in obigem Sinne) gilt:  $r_{\text{WF}}(x) = \frac{m_{\text{WF}}(x)}{m^*(x)} < 2$

**Aufgabe 6** (Reduktionsmethode für co-NPC-Sprachen)

[17 Punkte]

Betrachten Sie die folgenden zwei Sprachen:

$$\begin{aligned}\text{UNSAT} &:= \{\langle \varphi \rangle \in L_{\text{KNF}} \mid \forall \text{ Belegungen } v \text{ gilt: } v(\varphi) = 0\} \\ \text{DNF-TAUT} &:= \{\langle \varphi \rangle \in L_{\text{DNF}} \mid \forall \text{ Belegungen } v \text{ gilt: } v(\varphi) = 1\}\end{aligned}$$

wobei  $L_{\text{KNF}}$  bzw.  $L_{\text{DNF}}$  die Sprache der (codierten) *KNF-Formeln* bzw. *DNF-Formeln* bezeichne.

Zeigen Sie:

- (a) die Implikation „ $\implies$ “ der Aussage  $(L \subseteq \Sigma^*)$ :  $L \in \text{co-NPC} \iff \bar{L} \in \text{NPC}$
- (b)  $\text{UNSAT} \in \text{co-NPC}$
- (c)  $\text{UNSAT} \leq_p \text{DNF-TAUT}$
- (d)  $\text{DNF-TAUT} \in \text{co-NPC}$

**Hinweis:** Sie dürfen verwenden, daß man den (Wahrheits-) Wert  $v(\varphi)$  einer DNF-Formel unter einer gegebenen Belegung  $v$  in Zeit  $O(n^2)$  berechnen kann.

**Aufgabe 7** (Reduktionsmethode für NPC-Sprachen)

[11 Punkte]

Betrachten Sie die 3-KNF-Formel

$$\varphi \equiv (x_1 \vee x_4 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_3 \vee x_4) \wedge (\bar{x}_3 \vee \bar{x}_1 \vee \bar{x}_2)$$

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

- (a) Geben Sie den gemäß Vorlesung zu  $\varphi$  konstruierten gerichteten Graphen  $G_\varphi$  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  $x_j$ -Wege und  $\bar{x}_j$ -Wege.

- (b) Kennzeichnen Sie in  $G_\varphi$  (mit geeigneter Farbe) den gemäß Vorlesung bestimmten Hamiltonkreis, der zur erfüllenden Belegung  $v$  von  $\varphi$  gehört.

**Viel Erfolg!**