



Klausur Komplexitätstheorie WS 00/01
 13. Februar 2001

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:

Klausurcode: 00000000
Bitte notieren!

abgegeben: 2 Aufgabenblätter
 ... eigene Blätter

Aufgabe	1	2	3	4	5	6	7	8	Ges.
erreichbare Punktzahl	30	12	8	4	8	9	14	15	100
erreichte Punktzahl									

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

Aufgabe 1

[30 Punkte]

Bitte kreuzen Sie für jede der folgenden 10 Fragen entweder „JA“ oder „NEIN“ in jeder Zeile 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, kein Punkt.

Hinweis: Bisher nur vermutete oder bisher unbekannte Aussagen sind zu verneinen.

- (a) Sei $L' \in \text{DTIME}(n^2)$. Sei L eine Sprache und f eine Funktion, für die $x \in L \Leftrightarrow f(x) \in L'$ gilt. Welche der folgende Aussagen ist/sind *bewiesenermaßen* korrekt?
 JA NEIN
- Wenn f auf einer $O(n)$ -zeitbeschränkten Turingmaschine berechenbar ist, dann gilt $L \in \text{DTIME}(n^2)$.
 - Wenn f auf einer $O(n^2)$ -zeitbeschränkten Turingmaschine berechenbar ist, dann gilt $L \in \text{DTIME}(n^2)$.
 - Wenn f auf einer $O(2^n)$ -zeitbeschränkten Turingmaschine berechenbar ist, dann gilt $L \in \text{DTIME}(n \cdot 2^n)$.
- (b) Welche der folgende Aussagen ist/sind für alle Sprachen L und L' korrekt?
 JA NEIN
- Wenn $L \in \text{NP}$ und $L \leq_p L'$ gilt, so auch $L' \in \text{NP}$.
 - Wenn $L \in \text{NP}$ und $L' \leq_p L$ gilt, so auch $L' \in \text{NP}$.
 - Wenn $L \in \text{NPC}$ und $L' \leq_p \text{CLIQUE}$ gilt, so auch $L' \leq_p L$.
- (c) Welche der folgenden Aussagen ist/sind *bewiesenermaßen* korrekt?
 JA NEIN
- Für $k \geq 3$ gilt: $\{\langle G \rangle \mid G \text{ ist ein Graph, der eine Clique der Größe } k \text{ besitzt}\} \in \text{P}$.
 - Für $k \geq 3$ gilt: $k\text{-COLORING} \in \text{P}$.
 - Es gibt ein $k \in \mathbb{N}$, für das $k\text{-COLORING} \in \text{P}$ gilt.

- (d) Welche der folgenden Aussagen ist/sind für alle nichtdeterministischen Turingmaschinen M bewiesenermaßen richtig oder stimmen per Definition?

JA NEIN

Es gilt $L_M \in \text{NP}$.

Wenn $L_M \in \text{NP}$ gilt, dann ist L_M NP-schwer.

Wenn M $O(f(n))$ -zeitbeschränkt ist, dann gibt es eine deterministische Turingmaschine M' mit $L_M = L_{M'}$, die $O(f(n) \cdot 2^{f(n)})$ -zeitbeschränkt ist.

- (e) Der Satz von Cook sagt aus, daß...

JA NEIN

SAT $\in \text{NP}$ gilt und sich jede Sprache $L \in \text{NP}$ in polynomialer Zeit auf SAT reduzieren läßt.

es eine nichtdeterministische Turingmaschine M gibt, die das Erfüllbarkeitsproblem für KNF-Formeln entscheidet.

das Erfüllbarkeitsproblem für KNF-Formeln nicht in polynomialer Zeit entscheidbar ist.

- (f) Aus $\text{P} \neq \text{NP}$ folgt, daß...

JA NEIN

es keinen Polynomialzeitalgorithmus gibt, der INDEPENDENT-SET entscheidet.

für alle $L \in \text{NP}$ gilt: $L \notin \text{P}$.

es keinen polynomialen Approximationsalgorithmus für TRAVELING-SALESPERSON mit worst-case-Güte 8 gibt.

- (g) Es ist ein offenes Problem, ob...

JA NEIN

es für jede polynomialzeitbeschränkte NTM M eine polynomialzeitbeschränkte DTM M' mit $L_M = L_{M'}$ gibt.

für alle Sprachen $L \in \text{P}$ auch $\bar{L} \in \text{P}$ gilt.

co-NP = NP gilt.

3-SAT \leq_p INDEPENDENT-SET gilt.

es ein Polynom p gibt, so daß für alle NTM M , die SAT entscheiden, die Höhe des Berechnungsbaumes $CT_M(x)$ durch $p(|x|)$ beschränkt ist.

es einen Polynomialzeitalgorithmus gibt, der VERTEX-COVER entscheidet.

- (h) Welche der folgenden Aussagen ist/sind für alle $L \in \text{P}$ richtig?

JA NEIN

$\{x \in \{0, 1\}^* \mid \forall y \in \{0, 1\}^{|x|} : x\#y \in L\} \in \text{co-NP}$

$\{x \in \{0, 1\}^* \mid \exists y \in \{0, 1\}^{|x|} \exists z \in \{0, 1\}^{|x|} : x\#yz \in L\} \in \text{NP}$

$\{x \in \{0, 1\}^* \mid \exists y_1 \in \{0, 1\}^{|x|} \forall y_2 \in \{0, 1\}^{|x|} \exists y_3 \in \{0, 1\}^{|x|} : x\#y_1\#y_2\#y_3 \in L\} \in \Sigma_3^P$

- (i) Welche der folgenden Aussagen ist/sind für jede Sprache L korrekt?

JA NEIN

Aus 3-COLORING $\leq_p L$ und $L \leq_p$ 3-COLORING folgt $L \in \text{NPC}$.

Wenn L NP-schwer ist, so gilt 3-COLORING $\leq_p L$.

Wenn 3-COLORING $\leq_p L$ gilt, so ist L NP-schwer.

Aufgabe 2 (Definitionen aus der Vorlesung)

[12 Punkte]

Geben Sie die Definition der folgenden Konzepte bzw. Abkürzungen aus der Vorlesung an:

- (a) $L \leq_p L'$
- (b) L ist NP-vollständig
- (c) $\text{NTIME}(t(n))$
- (d) Die Sprache BINPACKING
- (e) Die Sprache TAUTOLOGIE
- (f) Entscheidungsvariante eines NPO-Problems
- (g) co-NP
- (h) $\text{NSPACE}(\log n)$

Aufgabe 3 (Resultate aus der Vorlesung)

[8 Punkte]

Zeigen Sie:

- (a) $\text{ELEMENT-UNIQUENESS} \leq_p \text{SORTIEREN}$
- (b) Formulieren und beweisen Sie die sich ergebende untere Schranke für SORTIEREN.

Aufgabe 4 (Resultate aus der Vorlesung)

[4 Punkte]

Geben Sie die „Reduktionsfunktion“ für die Reduktion $\text{RUCKSACK}^* \leq_p \text{PARTITION}$ an und formulieren Sie die Behauptungen, die über diese Funktion bewiesen werden müssen.

Aufgabe 5 (Resultate aus der Vorlesung)

[8 Punkte]

Zeigen Sie: Aus der Annahme $\text{P} \neq \text{NP}$ folgt, daß das Problem $\text{MINIMUM-TRAVELING-SALESPERSON}$ keinen asymptotischen Approximationsalgorithmus endlicher Güte besitzt, d.h.

es gibt keine Konstante k , zu der ein Polynomialzeitalgorithmus \mathcal{A} existiert, der auf jeder Eingabe x für $\text{MINIMUM-TRAVELING-SALESPERSON}$ mit Wert (Länge der kürzesten Rundreise) $\geq N$ stets eine Rundreise π mit $m(x, \pi) \leq k \cdot N$ ausgibt.

Aufgabe 6 (Charakterisierung von NP-Sprachen)

[9 Punkte]

Betrachten Sie die folgende Sprache

$$\text{GRAPH-COLORING} := \{\langle G \rangle \text{bin}(k) \mid k \in \mathbb{N}, G \text{ ist ein } k\text{-färbbarer Graph}\}$$

Geben Sie eine Sprache $L_0 \in \text{P}$ über einem geeigneten Alphabet Σ und ein Polynom p an, so daß für alle $x \in \{0, 1\}^*$ gilt:

$$x \in \text{GRAPH-COLORING} \Leftrightarrow \exists y \in \{0, 1\}^{p(|x|)} : x \# y \in L_0$$

Aufgabe 7 (Güte von Approximationsalgorithmen)

[14 Punkte]

Betrachten Sie den folgenden Algorithmus WORST FIT DECREASING, kurz WFD, für das BINPACKING-Problem, wobei die Kapazität der Behälter $b = 1$ sei und die Behälter mit $1, 2, \dots$ bezeichnet sind:

Eingabe: $x := (a_1, \dots, a_n) \in \mathbb{Q}^n$, wobei $n \geq 1$ und $0 < a_i \leq 1$ für $1 \leq i \leq n$
 Sortiere die Folge (a_1, \dots, a_n) absteigend. Die so entstandene Folge sei (b_1, \dots, b_n) .
for $i := 1, 2, \dots, n$ **do**
 Packe b_i in einen minimal beladenen, angefangenen Behälter, der noch genügend Platz bietet, falls ein solcher existiert; andernfalls packe b_i in den ersten unbenutzten Behälter j .
end for
Ausgabe: $\text{WFD}(x) :=$ Anzahl der benutzten Behälter

Bezeichne $\text{opt}(x)$ die Anzahl der benutzten Behälter in einer optimalen Verteilung der Objekte einer Eingabe x auf die Behälter.

Zeigen Sie: Für alle Eingaben x gilt $\frac{\text{WFD}(x)}{\text{opt}(x)} < 2$.

Aufgabe 8 (Polynomialzeitreduktionen)

[15 Punkte]

Das Problem MINIMUM-SET-COVER, informal beschrieben, besteht in der folgenden Aufgabe: Gegeben ist eine Matrix $A := (a_{ij})_{i,j} \in \{0, 1\}^{m \times n}$, in der jede Spalte mindestens eine 1 enthält. Gesucht ist eine Auswahl von möglichst wenigen Zeilen derart, daß in jeder Spalte der Matrix A' eine 1 steht, wobei A' aus A entsteht, indem man alle nicht ausgewählten Zeilen streicht. Wir formulieren hier das Problem MINIMUM-SET-COVER in zwei Varianten:

Variante 1 (*Parameteroptimierung*) Zu einer Matrix $A := (a_{ij})_{i,j} \in \{0, 1\}^{m \times n}$, in der jede Spalte mindestens eine 1 enthält, bestimme das *kleinste* k , für das eine Teilmenge $I \subseteq \{1, \dots, n\}$ mit $|I| = k$ existiert, die

$$(*) \quad \forall j \in \{1, \dots, n\} \exists i \in I: a_{ij} = 1$$

erfüllt.

Variante 2 (*Suchproblem nach optimaler Struktur*) Gegeben sei eine Matrix $A := (a_{ij})_{i,j} \in \{0, 1\}^{m \times n}$, in der jede Spalte mindestens eine 1 enthält. Gesucht ist eine Teilmenge $I \subseteq \{1, \dots, n\}$, die (*) erfüllt und für die $|I|$ *minimal* ist, d.h. für jede andere Auswahl I' , die (*) erfüllt, gilt $|I| \leq |I'|$.

Geben Sie einen Algorithmus \mathcal{A}_2 an, der Variante 2 unter Verwendung eines Algorithmus \mathcal{A}_1 für Variante 1 löst. Die Laufzeit von \mathcal{A}_2 ohne die Aufrufe für \mathcal{A}_1 soll polynomiell in der Länge der Eingabe sein.

Viel Erfolg!