Prof. Martin Hofmann
Steffen Jost

Ludwig-Maximilians-Universität München
Institut für Informatik

Wintersemester 2002/03
7. November 2002

Übung 3

Komplexitätstheorie

Aufgabe G11: Die Gültigkeit folgender Aussagen ist unbekannt:

a) $\mathcal{P} = co\mathcal{NP}$
b) $\mathcal{P} = \mathcal{NP} \cap co\mathcal{NP}$
c) $\mathcal{NP} \neq co\mathcal{NP}$
d) $\emptyset \neq \mathcal{NP} \cap co\mathcal{NP}$-vollständig

Finde möglichst viele Implikationen zur Aussage $\mathcal{P} = \mathcal{NP}$!

Aufgabe G12: Eine Sprache L ist *selbst-reduzierend*, falls sie von einer deterministischen, polynomial zeitbeschränkten Turing Maschine T erkannt wird, welche bei einer Eingabe der Länge n zusätzlich ein Orakel für $L_{<n}$ befragen darf. $L_{<n}$ bezeichne die Sprache aller Wörter aus L mit einer Länge kleiner als n.

a) Zeige das SAT selbst-reduzierend ist!

b) Diskutiere, ob damit bereits jede Sprache aus $\mathcal{NP}$ selbst-reduzierend ist!

Aufgabe G13:

a) Gebe eine konstruktive Aufzählung der Sprachen in $\mathcal{P}$ an!
Hinweis: Nach der Definition genügt es, eine totale berechenbare Funktion von den natürlichen Zahlen in die Menge aller deterministischen Turing Maschinen anzugeben, so dass jede Sprache aus $\mathcal{P}$ von mindestens einer der aufgezählten Maschinen erkannt wird und umgekehrt jede erkannte Sprache in $\mathcal{P}$ ist. Die genaue Laufzeit dieser Maschinen ist dabei ohne Belang, sie müssen jedoch bei jeder Eingabe terminieren.

b) Gebe eine konstruktive Aufzählung der Sprachen in $\mathcal{NP}$ an!

c) Zeige, dass eine Sprache aus $\mathcal{NP}$ bereits $\mathcal{NP}$-vollständig ist, wenn sie sich nur auf endlich vielen Wörtern von der Sprache SAT unterscheidet.
Bemerkung: Es sei an Aufgabe G6 a) erinnert.

d) Zeige, dass die Menge aller $\mathcal{NP}$-vollständigen Sprachen konstruktiv aufzählbar ist! Als Hilfe sei eine Aufzählung aller deterministisch polynomiell berechnbarer Funktionen gegeben (Die Konstruktion einer solche Aufzählung verläuft analog zu a).
Hinweis: Gebe eine Aufzählung deterministischer Turing Maschinen T_i an, so daß für alle i gilt: $L(T_i) \in \mathcal{NP}$-vollständig oder $L(T_i)$ unterscheidet sich von SAT nur an endlich vielen Stellen.

Korrektur zu H2:

- A, A_1, A_2 sollen endliche Listen natürlicher Zahlen sein, dürfen also auch Zahlen mehrfach enthalten.
- Zu zeigen ist, dass Partitionieren $\mathcal{NP}$-vollständig ist!

Korrekturen zu den Übungsblättern werden auch ständig auf der Webseite bekanntgegeben:
`http://www.tcs.informatik.uni-muenchen.de/lehre/WS02-03/TCS/#Aufgaben`