Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Ernst W. Mayr

WS 2001/02
Übungsblatt 12
18. Januar 2002

Komplexitätstheorie

Abgabetermin: Freitag, 25. Januar 2002, in der Vorlesung

Aufgabe 1

Die Sprache K, die in der Vorlesung definiert wurde, kann für eine beliebige Sprache $L \subseteq \Sigma^*$ wie folgt relativiert werden:

$$K(L) = \left\{ \langle 1^t, x, N \rangle \;\middle|\; \begin{array}{l} \text{Die nichtdeterministische Orakel-TM } N \text{ akzeptiert} \\ x \text{ mit Orakel } L \text{ in höchstens } t \text{ Schritten} \end{array} \right\}$$

Weiter kann man definieren: $K_1 = K$, und für alle $i \geq 2 : K_i = K(K_{i-1})$. Zeigen Sie, dass für alle $i \geq 1$ gilt: K_i ist many-one vollständig für Σ_i.

Aufgabe 2

Einleitung/Motivation: Wie in der Vorlesung besprochen wurde, folgt aus $\Sigma_i = \Sigma_j$ für $0 \leq i \leq j$, dass $\Sigma_i = \Sigma_k$ für alle natürlichen Zahlen k mit $k \geq i$, d.h., “die Polynomielle Hierarchie kollabiert auf die i-te Stufe”. Weiterhin wurde kurz angesprochen, dass man annimmt (aber noch nicht beweisen konnte), dass die Polynomielle Hierarchie unendlich ist, d.h. ein Kollaps auf die i-te Stufe wird für unwahrscheinlich gehalten (ähnlich der Tatsache, dass man $\mathcal{NP} = \mathcal{P}$ für unwahrscheinlich hält). Je kleiner dabei i ist, für desto unwahrscheinlicher hält man einen solchen Kollaps.

Es ist in der Komplexitätstheorie grundlegend, dass man die $\mathcal{NP}$-vollständigen Probleme als schwerer zu entscheiden erachtet als die Probleme in $\mathcal{P}$. Es wäre interessant, ähnliche Unterscheidungen zwischen leichteren und schwereren Problemen auch für andere Teilmengen von $\mathcal{NP}$ vorzunehmen. Dazu ist die folgende Beobachtung interessant: Falls es eine Sprache gibt, die in $\mathcal{NP} \cap$ co-$\mathcal{NP}$ liegt und gleichzeitig $\mathcal{NP}$-vollständig ist, dann kollabiert die Polynomielle Hierarchie auf die erste Stufe. Da man dies als unwahrscheinlich erachtet (s. oben), kommt man zu folgender Aussage: “Die Probleme in $\mathcal{NP} \cap$ co-$\mathcal{NP}$ sind wahrscheinlich in einem bestimmten Sinn leichter zu entscheiden als die $\mathcal{NP}$-vollständigen Probleme”. Das bedeutet, dass beispielsweise das Problem Factoring, das auf dem letzten Übungsblatt eingeführt wurde, wahrscheinlich in einem bestimmten Sinn leichter zu entscheiden ist als Sat.

Es wäre schön, diese Beobachtung systematisch auf Probleme in $\mathcal{NP}$, von denen man nicht weiß, ob sie in $\mathcal{NP} \cap$co-$\mathcal{NP}$ liegen, erweitern zu können. Dazu verwendet man eine Struktur, die mit Low Hierarchy und High Hierarchy bezeichnet wird. Wir wollen im Folgenden die Low und die High Hierarchy definieren und einige einfache Eigenschaften der Low und der High Hierarchy beweisen.

Definition: Man definiert für $i \geq 0$ die i-te Stufe L_i und H_i der Low und der High Hierarchy als

$$L_i = \{L \in \mathcal{NP};\ \Sigma_i(L) \subseteq \Sigma_i\}$$

$$\text{und} \quad H_i = \{L \in \mathcal{NP};\ \Sigma_{i+1} \subseteq \Sigma_i(L)\}.$$

Zeigen Sie nun die folgenden Eigenschaften der Low und der High Hierarchy:

1. $L_0 \subseteq L_1 \subseteq L_2 \subseteq L_3 \subseteq \ldots$ und $H_0 \subseteq H_1 \subseteq H_2 \subseteq H_3 \subseteq \ldots$
2. Falls für ein $i \geq 0$ gilt, dass $L_i \cap H_i \neq \emptyset$, dann kollabiert die Polynomielle Hierarchie auf die i-te Stufe.

Aufgabe 3

Zeigen Sie die folgenden weiteren Eigenschaften der Low und der High Hierarchy:

1. $L_0 = \mathcal{P}$, $H_0 = \{L \in \mathcal{NP};\ \forall L' \in \mathcal{NP} : L' \leq_T L\}$.
2. $L_1 = \mathcal{NP} \cap \text{co-}\mathcal{NP}$, $H_1 = \{L \in \mathcal{NP};\ \forall L' \in \mathcal{NP} : L' \leq_{SN} L\}$. Dabei bezeichnet $\leq_{SN}$ die in der Vorlesung eingeführte starke nichtdeterministische Reduzierbarkeit.

Anmerkung: Die Low und die High Hierarchy sind ein nützliches Tool zur Klassifizierung von Sprachen; so konnte man zeigen, dass das Graphenisomorphieproblem GI in L_2 enthalten ist, obwohl man Ende der siebziger Jahre noch spekuliert hatte, dass GI $\mathcal{NP}$-vollständig sein könnte. Damit weiß man jetzt also, dass diese Spekulation nur dann eintreten kann, wenn die Polynomielle Hierarchie auf die zweite Stufe kollabiert; damit ist sie sehr unwahrscheinlich. Ebenso konnte man zeigen, dass alle Tally-Sprachen und alle dünnen Sprachen in L_2 enthalten sind, und dadurch ebenfalls den Schwierigkeitsgrad dieser Sprachen als relativ niedrig einordnen.

Aufgabe 4

Zeigen Sie dass gilt: Falls $\mathcal{NP} \subseteq BPP$, dann ist $R = \mathcal{NP}$.
Hinweis: Falls $\mathcal{NP} \subseteq BPP$, dann ist Sat in BPP. Erinnern Sie sich, dass Sat selbstreduzierbar ist, und benutzen Sie diese Eigenschaft, um einen BPP-Algorithmus zu konstruieren, der für eine gegebene Formel mit hoher Wahrscheinlichkeit eine erfüllende Belegung findet, falls eine solche existiert. Testen Sie anschließend, ob die gegebene Formel durch die gefundene Belegung tatsächlich erfüllt ist, um einen R-Algorithmus für Sat zu erhalten.