AG Theorie der künstlichen Intelligenz
FB Mathematik und Informatik, Universität Bremen
Prof. Dr. Carsten Lutz, Dr. Thomas Schneider

Cartesium 2.59, 2.56
{clu,schneidt}@uni-bremen.de
Tel.: 0421/218-64431, -64432

6. Aufgabenblatt für die Vorlesung „Komplexitätstheorie“

Aufgabe 1: 25%

Finde eine *entscheidbare* Sprache in $\mathsf{P}_{/\mathsf{poly}}$, die nicht in P ist. Begründe Deine Wahl.

Aufgabe 2: 25%

Das *Erfüllbarkeitsproblem für Schaltkreise (Circuit SAT, CSAT)* ist

$$\text{CSAT} := \{C \mid C \text{ Schaltkreis, der für mindestens eine Eingabe den Ausgabewert 1 liefert}\}.$$

Zeige, dass CSAT NP-hart ist, ohne die NP-Härte eines anderen Problemes wie SAT oder 3SAT zu verwenden. Verwende dazu die Definition von NP (also polynomielle Beweissysteme) und die Schaltkreise, die im Beweis des Theorems auf Folie 15 in Kapitel 6 („$\mathsf{P} \subseteq \text{uniform } \mathsf{P}_{/\mathsf{poly}}$“) konstruiert wurden.

Zeige danach, dass CSAT $\leq_p$ SAT. Wir haben also einen alternativen NP-Härte-Beweis für SAT gefunden.

Aufgabe 3: 25%

Eine *Linearzeitreduktion* ist eine Reduktion, die in Zeit $\mathcal{O}(n)$ berechnet werden kann. P-*Härte und* P-*Vollständigkeit bezüglich Linearzeitreduktionen* sind dann in der offensichtlichen Weise definiert (LogSpace-Reduktionen werden ausgetauscht durch Linearzeit-Reduktionen). Verwende das Hierarchietheorem, um zu zeigen, dass es keine P-vollständigen Probleme bezüglich Linearzeitreduktionen gibt.

Aufgabe 4: 25%

Zeige, dass ein Kollaps der polynomiellen Hierarchie auch bereits durch den Kollaps einer Ebene verursacht wird: wenn $\Sigma_k^p = \Pi_k^p$, dann $\mathsf{PH} = \Sigma_k^p$. Hinweis: verwende das Lemma auf Folie 13 in Kapitel 7 („wenn $\Sigma_k^p = \Sigma_{k+1}^p$, dann $\mathsf{PH} = \Sigma_k^p$ “) sowie die Charakterisierung von Σ_k^p auf Folie 16.

Aufgabe 5: 25% (Zusatzaufgabe)

Linear programming (LP) ist das Problem, zu entscheiden, ob ein System linearer Ungleichungen der Form

$$c_1 \cdot x_1 + \cdots + c_n \cdot x_n = c \text{ und } c_1 \cdot x_1 + \cdots + c_n \cdot x_n \leq c$$

eine Lösung in den nicht-negativen rationalen Zahlen hat. Zeige durch Reduktion von CVP, dass LP P-hart ist.

Hinweis: Gehe ähnlich vor wie beim Beweis der NP-Härte von Integer Programming. In LP kann man im allgemeinen nicht durch Gleichungen sicherstellen, dass eine numerische Variable nur den Wert 0 oder 1 hat! Das ist jedoch auch nicht notwendig: entwirf die Gleichungen so, dass alle numerischen Variablen den Wert 0 oder 1 annehmen müssen, *weil alle Eingaben entweder 0 oder 1 als Wert haben.*