

Komplexitätstheorie

Übungsblatt 6

Abgabe als PDF bis 3. 7. 2018, 23:59 Uhr in Stud.IP, Ordner „Abgabe Übungsblatt 6“
 Bitte nur eine PDF-Datei pro Gruppe, Lizenz „Selbst verfasstes, nicht publiziertes Werk“.

1. (25 %) Finde eine *entscheidbare* Sprache in $P_{/poly}$, die nicht in P ist. Begründe Deine Wahl.

2. (25 %) Das *Erfüllbarkeitsproblem für Schaltkreise* ist

$CSAT := \{C \mid C \text{ Schaltkreis mit } C(w) = 1 \text{ für mindestens eine Eingabe } w\}$.

a) Zeige, dass $CSAT$ NP-hart ist, ohne die NP-Härte eines anderen Problems wie SAT oder $3SAT$ zu verwenden. Verwende dazu die Definition von NP (mittels polynomieller Beweissysteme) und die Schaltkreise, die im Beweis von Theorem 6.13 („ $P \subseteq \text{Uniform-}P_{/poly}$ “) konstruiert wurden.

b) Zeige $CSAT \leq_p SAT$.

(Dies ergibt also einen alternativen NP-Härte-Beweis für SAT .)

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 ersetzt durch Linearzeit-Reduktionen). Verwende das Hierarchietheorem, um zu zeigen, dass es keine P-vollständigen Probleme bezüglich Linearzeitreduktionen gibt.

4. (25 %) Zeige, dass ein Kollaps der polynomiellen Hierarchie auch bereits durch den Kollaps einer Ebene verursacht wird, d. h.: wenn $\Sigma_k^p = \Pi_k^p$, dann $PH = \Sigma_k^p$.

Hinweis: verwende Lemma 7.9 („wenn $\Sigma_k^p = \Sigma_{k+1}^p$, dann $PH = \Sigma_k^p$ “) sowie die Charakterisierung von Σ_k^p mittels Π_{k-1}^p -Beweissystemen (Lemma 7.14).

5. **Zusatzaufgabe** (20 %) *Linear programming* (LP) ist das Problem zu entscheiden, ob ein System linearer Ungleichungen der Form

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

eine Lösung in den nichtnegativen 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*.