Übungen zur Vorlesung Komplexitätstheorie

WWU Münster
Institut für Informatik

Prof. Dr. Markus Müller-Olm
Sebastian Kenter

SS 2016 — **Übungsblatt 7** — 09.06.2016

Abgabe: *In Dreiergruppen*, bis Donnerstag, 16.06.16, vor der Vorlesung in BK 61.

Besprechung: Die Aufgaben werden in der Übung am Mittwoch, dem 22.06.16, um 10:15 Uhr besprochen.

Aufgabe 7.1. **[Verifizierer mit logarithmischem Platzbedarf]** (10 Punkte)
Eine *Turingmaschine mit Zertifikatband* ist eine Turingmaschine mit einem Eingabeband zur Aufnahme einer Eingabe x, einem Zertifikatband zur Aufnahme eines Zertifikats u und k Arbeitsbändern, wobei die Köpfe auf Eingabe- und Zertifikatband nicht schreiben, sondern nur lesen und nach links und rechts bewegt werden können.

Zeigen Sie: Eine Sprache $L \subseteq \{0,1\}^*$ liegt genau dann in NP, wenn es ein Polynom p und eine Turingmaschine M mit Zertifikatband gibt, die auf jedem ihrer Arbeitsbänder höchstens $O(log\, n)$ viele Felder in Abhängigkeit von der Eingabelänge n besucht, derart dass für alle $x \in \{0,1\}^*$ gilt

$$x \in L \iff \exists u \in \{0,1\}^{p(|x|)} : M(x,u) = 1.$$

Aufgabe 7.2. **[Mehrfach exponentieller Zeit- oder Platzbedarf]** (10 Punkte)
Wir definieren zunächst induktiv $Tow(0, n^c) = n^c$ und $Tow(k{+}1, n^c) = 2^{Tow(k,n^c)}$. Betrachten Sie nun die folgenden Komplexitätsklassen:

$$\begin{aligned} k\mathsf{EXP} &= \cup\{\mathrm{DTIME}(Tow(k,n^c)) | c \geq 1\}, \\ k\mathsf{NEXP} &= \cup\{\mathrm{NTIME}(Tow(k,n^c)) | c \geq 1\}, \\ k\mathsf{EXPSPACE} &= \cup\{\mathrm{SPACE}(Tow(k,n^c)) | c \geq 1\}, \\ k\mathsf{NEXPSPACE} &= \cup\{\mathrm{NSPACE}(Tow(k,n^c)) | c \geq 1\}. \end{aligned}$$

$2\mathsf{EXP}$ und $k\mathsf{EXP}$ nennt man auch die Klassen der *Sprachen mit doppelt bzw. k-fach exponentiellem Zeitbedarf*, $2\mathsf{EXPSPACE}$ und $k\mathsf{EXPSPACE}$ die der *Sprachen mit doppelt bzw. k-fach exponentiellem Platzbedarf*. Zeigen Sie:

a) $0\mathsf{EXP} = \mathsf{P}, 0\mathsf{NEXP} = \mathsf{NP}, 0\mathsf{EXPSPACE} = \mathsf{PSPACE}, 1\mathsf{EXP} = \mathsf{EXP}, 1\mathsf{NEXP} = \mathsf{NEXP}$.

b) Für alle $k \geq 0$ gilt $k\mathsf{EXP} \subseteq k\mathsf{NEXP} \subseteq k\mathsf{EXPSPACE} = k\mathsf{NEXPSPACE} \subseteq (k+1)\mathsf{EXP}$.

Aufgabe 7.3. **[NL-harte Sprachen]** (10 Punkte)
Zeigen Sie, dass jede nicht-triviale Sprache NL-hart relativ zu Polynomzeit-Reduktionen ist, d.h. dass sich jede Sprache aus NL auf sie Polynomzeit-reduzieren lässt.