Hans U. Simon
Malte Darnstädt

Bochum, den 21.04.2010
Abgabe bis zum 28.04. in der Übung

Übungen zur Vorlesung

Komplexitätstheorie

Sommer 2010

Übungsblatt 1

Aufgabe 1.1

Gib eine Kodierung für beliebige Turing-Maschinen (beliebig viele Zustände und Bänder, beliebiges Alphabet) an, die nur Zeichen aus dem Alphabet $\{0, 1\}$ verwendet. Jede Turing-Maschine soll unendlich viele Kodierungen besitzen.
Es muss möglich sein in polynomieller Zeit aus der Kodierung die ursprüngliche TM (oder eine sich gleich verhaltende TM) zu rekonstruieren.

Aufgabe 1.2

Zeige, dass eine Turing-Maschine existiert, die bei einer Eingabe x der Länge $n := |x|$ in $O(\text{poly}(n))$-Zeit entscheidet, ob x eine nach deiner Lösung zu Aufgabe 1 kodierte Turing-Maschine darstellt.

Aufgabe 1.3

Eine *partielle Funktion* von $\{0, 1\}^*$ nach $\{0, 1\}^*$ ist eine Funktion, die nicht auf allen Eingaben definiert ist. Wir sagen, dass die Turing-Maschine M die partielle Funktion f berechnet, wenn für alle $x \in \{0, 1\}^*$ gilt:

- Falls f auf x definiert ist, gilt $M(x) = f(x)$
- Ansonsten geht M auf x gestartet in eine Endlosschleife über

Sei $\mathcal{F}$ die Menge der **berechenbaren**, partiellen Funktionen und $\mathcal{S} \subset \mathcal{F}$.
Definiere die Funktion $f_{\mathcal{S}} : \{0, 1\}^* \to \{0, 1\}$ wie folgt:

$$f_{\mathcal{S}}(\alpha) = \begin{cases} 1 & M_\alpha \text{ berechnet eine partielle Funktion aus } \mathcal{S} \\ 0 & \text{sonst} \end{cases}$$

Beweise den *Satz von Rice*:
Für alle $\mathcal{S} \neq \emptyset, \mathcal{S} \neq \mathcal{F}$ ist die Funktion $f_{\mathcal{S}}$ nicht berechenbar.

Aufgabe 1.4

Im Folgenden definieren wir eine einfache Programmiersprache:

Wir verfügen über ein einziges, unendlich großes Array A, das Elemente aus $\{0, 1, \Box\}$ speichern kann. Beim Programmstart befindet sich in den ersten n Feldern von A die Eingabe $x \in \{0, 1\}^n$, auf allen anderen Feldern steht das Zeichen $\Box$.

Außerdem haben wir einen Zähler i, der anfangs auf $i = 0$ gesetzt ist.

Ein Programm besteht aus Zeilen der Form:

`label`: `wenn` $A[i]$ `gleich` σ `dann` `befehle`

Wobei $\sigma \in \{0, 1, \Box\}$ und `befehle` eine Liste von mindestens einem der folgenden Befehle ist:

- `Setze` $A[i]$ `auf` τ (wobei $\tau \in \{0, 1, \Box\}$)
- `Gehe zu` `label` (anstatt die Befehlsliste weiter abzuarbeiten und dann mit der nächsten Zeile weiterzumachen, geht das Programm direkt in die Zeile `label`)
- `Erhöhe` i `um eins`
- `Verkleinere` i `um eins` (falls $i = 0$ bewirkt dieser Befehl nichts)
- `Gib` b `aus und halte an` (wobei $b \in \{0, 1\}$)

Das Programm wird von der ersten Zeile an abgearbeitet.

Beweise, dass für jede Funktion $f : \{0, 1\}^* \to \{0, 1\}$, die von einem Programm in Zeit $O(T(n))$ berechnet werden kann (wobei $T(n)$ zeitkonstruierbar ist), gilt:

$f \in \mathbf{DTIME}(T(n))$

Informationen zu den Übungen

- Die Übungen finden mittwochs um 14 Uhr (c.t.) in NA 2/64 statt
- Übungsgruppenleiter: Malte Darnstädt (NA 1/70)
Sprechstunde freitags von 11 bis 12 Uhr
- Korrektur der Übungsblätter: Florian Giesen (NA 5/71)
- Neue Übungsblätter werden in der Übung verteilt und sind mittwochnachmittags im Internet unter http://www.rub.de/lmi/lehre/kplx_ss10/ zu finden
- Auf jedem Übungsblatt gibt es vier Aufgaben mit jeweils vier erreichbaren Punkten
- Gruppen von bis zu drei Teilnehmern dürfen eine gemeinsame Lösung einreichen
- Die bearbeiteten Aufgaben sind am darauffolgenden Mittwoch in der Übung abzugeben. Eine vorherige Abgabe kann bei Florian (NA 5/71) oder Malte (NA 1/70) erfolgen
- Einen Übungsschein erhält, wer mindestens die Hälfte der Gesamtpunktzahl erreicht hat und in den Übungen eigene Lösungen vorgestellt hat