Entwurf und Analyse von Algorithmen

WS 1999/2000
Prof. Dr. K. Madlener
Übungsblatt 0

Aufgabe 0.1. Die am Seitenende angegebene Adresse (URL) führt auf unsere Vorlesungsseite im WWW.

a) Lesen Sie sich diese Seite bitte sorgfältig durch.
b) Fügen Sie ein Lesezeichen (bookmark) für diese Seite Ihrer Sammlung hinzu.
c) Tragen Sie sich in die auf der Seite beschriebenen Nachrichtenliste `eaa` ein, wenn Sie nicht bereits die Begrüßungsnachricht erhalten haben (dann sind Sie nämlich schon auf der Liste).

Aufgabe 0.2. Wir haben in der vergangenen Vorlesung Turingmaschinen mit einem zweiseitig-unendlichen Band eingeführt. Definieren Sie analog k-Band-Turingmaschinen, wobei $k \in \mathbb{N}^+$, also Turingmaschinen mit k Bändern. Dabei sollen der nächste Zustand, die neuen Inhalte der besuchten Bandfelder und die Kopfbewegungen vom aktuellen Zustand und von den aktuellen Inhalten der besuchten Bandfelder abhängen. Geben Sie dazu die Definition der k-Band-Turingmaschine als Tupel an, erklären Sie die einzelnen Bestandteile des Tupels, definieren Sie Konfigurationen und zeigen Sie, wie die Übergangsfunktion auf Konfigurationen wirkt. Definieren Sie auch die Begriffe „Rechnung", „die Maschine berechnet eine Funktion f" und „die Maschine entscheidet eine Sprache L".

Aufgabe 0.3. Geben Sie eine 1-Band-Turingmaschine an (siehe vergangene Vorlesung), die die Sprache $L = \{w \in \{a,b\}^* : w = w^{\mathrm{mi}}\}$ der Palindrome über $\{a,b\}$ entscheidet. Wenn $L_n = \{w \in L : |w| = n\}$ (für $n \in \mathbb{N}$) ist, wie lang, in Abhängigkeit von n, ist die längste Rechnung Ihrer Maschine für ein beliebiges $w \in L_n$? Versuchen Sie, eine Maschine anzugeben, deren längste Rechnung, in Abhängigkeit von n, ein Polynom höchstens zweiten Grades ist.

Geben Sie dann eine k-Band-Turingmaschine ($k \geq 2$) an, die L entscheidet und deren längste Rechnung, in Abhängigkeit von n, ein Polynom höchstens ersten Grades ist.

Aufgabe 0.4. Zeigen Sie, dass es für jede Sprache, die von einer k-Band-Turingmaschine ($k \geq 2$) entschieden wird, eine 1-Band-Turingmaschine gibt, die diese Sprache ebenfalls entscheidet.

Aufgabe 0.5. Wir wiederholen kurz die folgende Definition:

Sei φ ein Berechnungskonzept (Beispiele: eine universelle μ-rekursive Funktion, eine universelle Turingmaschine oder ein universelles WHILE-Programm). Ein Komplexitätsmaß für φ ist eine Abbildung $\Phi : \mathbb{N}^2 \to \mathbb{N}$ mit den Eigenschaften: (i) $\Phi(i,n)\downarrow$ genau dann, wenn $\varphi_i(n)\downarrow$ (für alle $i, n \in \mathbb{N}$) und (ii) die Menge $\{(i,n,m) \in \mathbb{N}^3 : \Phi(i,n) \leq m\}$ ist entscheidbar. Schreibweise: $\Phi_i(n) := \Phi(i,n)$.

Zeigen Sie nun, dass es für jedes Komplexitätsmaß Φ für φ und beliebige Funktionen $f \in \mathcal{R}^{(1)}$ und $h \in \mathcal{R}^{(1)}$ ein $i_0 \in \mathbb{N}$ gibt, so dass $f = \varphi_{i_0}$ und für alle $n \in \mathbb{N}$ gilt $\Phi_{i_0}(n) > h(n)$. Umgangssprachlich ausgedrückt: Es gibt beliebig ineffiziente Wege, eine berechenbare Funktion zu programmieren.

Informationen zur Vorlesung:
http://www-madlener.informatik.uni-kl.de/ag-madlener/lectures/eaa/ws1999-2000/eaa.html