Übungen zur Vorlesung

Komplexitätstheorie

Aufgabenblatt 3

In der Übung am 18.11.08 um 10.15 Uhr im HZ204
werden die Übungsaufgaben vorgerechnet.

Eine Turingmaschine M sei folgendes: $M = (Z, \Sigma, \Gamma, \delta, z_0, \Box, E)$, wobei

- Z ist endl. *Zustandsmenge.*
- Σ ist das *Eingabealphabet*
- $\Gamma \subset \Sigma$ ist das Arbeitsalphabet.
- $\delta : Z \times \Gamma \to \mathcal{P}(Z \times \Gamma \times \{L, R, N\})$
- $z_0 \in Z$ der *Startzustand.*
- $\Box \in \Gamma - \Sigma$ das *Leerzeichen.*
- $E \subseteq Z$ die *Endzustände.*
- besitzt k Arbeitsbänder.

Siehe Uwe Schöning: Theoretische Informatik - kurz gefaßt (S.80).

Aufgabe 1 (Bandkompression)
Skizzieren Sie, wie eine TM mit Platzverbrauch $2 \cdot f(|x|)$ (x ist Eingabe und f konstruierbar) durch eine äquivalente TM mit Platzverbrauch $f(|x|)$ simuliert werden kann. Wie verhält es sich mit dem Zeitbedarf?

Aufgabe 2 (Geklammerte Ausdrücke)
Zeige, dass die Sprache L, der wohlgeklammerten Ausdrücke über $\Sigma = \{(,)\}$ in Platz $O(\log n)$ entscheidbar ist. Es gilt $(()()) \in L$, aber $(()()(\notin L$.

Hinweis: Für die anzugebenden Konstruktionen ist nur eine Beschreibung auf abstraktem Niveau nötig und nicht jede Einzelheit von Interesse.

Aufgabe 3
Satz von Savitch
Sei $f : \mathbb{N}_0 \to \mathbb{N}$ konstruierbar und $\log \in O(f)$. Dann $NSPACE(f) \subseteq SPACE(f^2)$

Der Beweis geht wie folgt: Jede 1-Band-NTM M kann als gerichteten Konfigurationsgraph $G_M(x)$ für jede Eingabe x aufgefasst werden. Hierbei ist die Knotenmenge $V_M(x)$ eine Menge von Konfigurationen von M (Beschreibung von

Zustand und Bandinhalt): (q, y, z, p), q ist Zustand, y Bandinhalt links und unter Lese/Schreibkopf, z Bandinhalt rechts vom Lese/Schreibkopf und p Position des Lesekopfs in der Eingabe. Es gilt für die Kanten: $(a, b) \in A_M(x) \iff a$ kann mittels δ in b überführt werden. $Init_M(x)$ ist die Menge der Anfangskonfigurationen.
Ist $L \in NSPACE(f)$ und M eine NTM die L erkennt, dann besteht nun unsere Aufgabe darin, in $G_M(x)$ einen Weg von $Init_M(x)$ zu einer akzeptierenden Konfiguration zu suchen und zwar mit nicht mehr als $O(f^2)$ Platz. Dies bewerkstelligt folgender Algorithmus:

$acc \leftarrow$ **false**
for all $K \in V_M(x)$ **do**
    if K akzeptiert $\wedge$ $reach(Init_M(x), K, c'f(n))$ **then**
        $acc \leftarrow$ **true**
    end if
end for
return acc

function $reach(K_1, K_2, j)$:
if $j == 0$ **then**
    return $((K_1 == K_2) \vee (K_1 \xrightarrow{1} K_2))$
else
    $found \leftarrow$ **false**
    for all $K \in V_M(x)$ **do**
        if $reach(K_1, K, j-1) \wedge (K, K_2, j-1)$ **then**
            $found \leftarrow$ **true**
        end if
    end for
end if
return $found$

Hierbei liefert $reach(K_1, K_2, j)$ **true** zurück, falls $K_1 \overset{\leq 2^j}{\rightarrow} K_2$ gilt, sonst **false**. Die Konstante c' ist diejenige, so dass $T_M(x) \leq c'f(|x|)$ für alle Eingaben x.

Bestimme nun wieviel Platz und Zeit dieser Algorithmus asymptotisch braucht.

Aufgabe 4 (Linearer Speed-Up)
Sei $L \in TIME(f(n))$. Zeigen Sie: Für jedes $\epsilon > 0$ gilt $L \in TIME(f'(n))$ mit $f'(n) = \epsilon f(n) + n + 2$.
Tipp: Betachten Sie eine k-Band TM. Sei $m = \lceil \frac{6}{\epsilon} \rceil$. Sei $\Sigma' := \Sigma \cup \Sigma^m$. Fassen Sie jeweils m Zeichen der Eingabe in ein m-Tupel aus Σ' auf einem Extraband zusammen. Was für zusätzlich Zustände brauchen Sie hierfür.
Simulieren Sie nun auf dem kreierten m-Tupel-Wort die TM M, welche L in Zeit $f(n)$ erkennt.