Übungsblatt 1

Abgabe der schriftlichen Lösungen bis 30. Oktober 2008

Aufgabe 1 ***mündlich***
Betrachten Sie eine TSP-Instanz mit n Städten und (positiven) Entfernungen d_{ij}. Wir definieren für jede Menge S von Städten (mit $1 \notin S$) und für jedes $j \in S$ den Wert $c[S, j]$ als den kürzesten Weg von 1 nach j, der jede Stadt in $S \cup \{1\}$ genau einmal besucht.

(a) Geben Sie einen Algorithmus an, der alle $c[S, j]$ mit *dynamischer Programmierung* berechnet, also von kleinen auf große Stadtmengen S schließt.

(b) Benutzen Sie diesen Algorithmus, um das TSP in Zeit $O(n^2 2^n)$ zu lösen. Welchen Platzbedarf hat der resultierende Algorithmus?

Aufgabe 2 ***mündlich***
Betrachten Sie eine Turingmaschine M, die ein *zweidimensionales* Band zur Verfügung hat. Der Schreib-Lesekopf kann sich also auch nach oben und unten bewegen.

(a) Von welcher Form ist die Überführungsfunktion von M?

(b) Zeigen Sie, wie eine solche Turingmaschine durch eine DTM M' simuliert werden kann.

Aufgabe 3 ***mündlich***
Eine Turingmaschine heißt **blind** (engl. *oblivious*: vergesslich, blind) falls die Kopfpositionen zu jedem Zeitpunkt t ihrer Rechnung nur von t abhängen.

(a) Zeigen Sie, dass jede Turingmaschine M von einer blinden Turingmaschine M' simuliert werden kann.

(b) Geben Sie eine obere Schranke für die Rechenzeit $time_{M'}(x)$ von M' in Abhängigkeit von $time_M(x)$ an.

Aufgabe 4 ***mündlich***
Zeigen Sie: Jede $t(n)$-zeitbeschränkte k-DTM M kann von einer 1-DTM M' in Zeit $O(t(n)^2)$ simuliert werden. Lässt sich die Simulation von M bei Verwendung einer 2-DTM noch zeiteffizienter gestalten?

Aufgabe 5 ***mündlich***
Zeigen Sie: Jede Sprache, die von einer k-NTM in $f(n)$ vielen Schritten entschieden wird, kann auch von einer 2-NTM in $\mathcal{O}(f(n))$ vielen Schritten entschieden werden.

Aufgabe 6 ***10 Punkte***
Bestimmen Sie für alle Paare f, g der folgenden Funktionen, ob (a) $f(n) \in O(g(n))$, (b) $f(n) \in \Omega(g(n))$, oder ob (c) $f(n) \in \Theta(g(n))$ gilt:

(1) n^2, (2) n^3, (3) $n^2 \log n$, (4) 2^n, (5) n^n, (6) $n^{\log n}$,
(7) 2^{2^n}, (8) $2^{2^{n+1}}$, (9) n^2, falls n gerade ist, 2^n sonst.