14. Aufgabenblatt zur Vorlesung

Entwurf und Analyse von Algorithmen

(Günter Rote, Astrid Kaffanke, Christian Knauer)

Abgabe: 8. Februar 2001

1. DNF-SAT **und** SAT (5 Punkte).

   Zeige, dass das Erfüllbarkeitsproblem für aussagenlogische Formeln in *disjunktiver* Normalform DNF-SAT in polynomieller Zeit lösbar ist.

   Jede aussagenlogische Formel in konjunktiver Normalform φ_K läßt sich in eine äquivalente aussagenlogische Formel in disjunktiver Normalform φ_D überführen. Handelt es sich bei der Funktion $f : \varphi_K \mapsto \varphi_D$ um eine Polynomialzeitreduktion SAT $\leq_P$ DNF-SAT? Begründe Deine Antwort.

2. k-CLIQUE (4 Punkte).

   In der Übung habt ihr gezeigt, dass das Problem 2-Färbbarkeit in P liegt, und in der Vorlesung wurde bewiesen, dass k-Färbbarkeit für $k \geq 3$ NP-vollständig ist.

   Das Problem k-CLIQUE für eine feste Zahl $k \in \mathbf{N}$ ist folgendermaßen formuliert: Gegeben einen Graphen G, entscheide ob G eine Clique der Größe k hat.

   Kann man für k-CLIQUE abhängig von k ähnliche Aussagen wie für k-Färbbarkeit machen? Begründe Deine Antwort.

3. **Richtig oder Falsch** (1 Punkte).

   Ist die folgende Aussage richtig oder falsch? Begründe Deine Antwort. Falls sie falsch ist, formuliere eine ähnliche, aber richtige Aussage.

   Wenn $A \leq_P B$ und B lässt sich in linearer Zeit lösen, dann gibt es einen Linearzeit-Algorithmus für A.

4. ***Zusatzaufgabe*** **Entscheidungsproblem-Konstruktionsproblem** (4 zusätzliche Punkte).

   (a) Angenommen es gibt einen Algorithmus, der in polynomieller Zeit *entscheiden* kann, ob ein Graph einen Hamiltonkreis hat. Zeige, wie sich mit dessen Hilfe in polynomieller Zeit ein solcher Hamiltonkreis *konstruieren* (Das heisst, die Knoten sollen in zyklischer Reihenfolge angegeben werden) lässt.

   (b) Angenommen es gibt einen Algorithmus, der in polynomieller Zeit *entscheiden* kann, ob eine boolesche Formel eine erfüllende Belegung hat. Zeige, wie sich mit dessen Hilfe in polynomieller Zeit eine solche Belegung *konstruieren* lässt.

5. ***Zusatzaufgabe*** NP **und Turing-Maschine** (3 zusätzliche Punkte).

   Beweise, dass sich jedes Problem in NP mit einer deterministischen Turing-Maschine in exponentieller Zeit $O(2^{n^k})$ für ein bestimmtes k lösen lässt.