Theoretische Informatik II

Dr. Eva Richter / Holger Arnold
Universität Potsdam, Theoretische Informatik, Sommersemester 2008

Übungsblatt 11 (Version 3) — Abgabetermin: 14.7.2008, 12.00 Uhr

---

Aufgabe 11.1

Ein Homomorphismus h heißt *nichtlöschend*, wenn h kein Zeichen auf das leere Wort abbildet. Beweisen Sie, dass NP unter nichtlöschenden Homomorphismen abgeschlossen ist.

Aufgabe 11.2

Betrachten Sie das folgende (bereits aus der Vorlesung bekannte) Problem:

$$\begin{aligned} \mathit{SUBSET\text{-}SUM} = \{\langle S, t\rangle \mid S = \{x_1, \ldots, x_k\} \subset \mathbb{N} \text{ und } \sum_{i=1}^{l} y_i = t \\ \text{für eine Teilmenge } \{y_1, \ldots, y_l\} \subseteq S\}. \end{aligned}$$

In einem Wort $\langle S, t\rangle \in \mathit{SUBSET\text{-}SUM}$ seien die Zahlen in S und die Zahl t binär codiert. Sei f die wie folgt definierte (offensichtlich total berechenbare) Abbildung:

(i) Sei ϕ eine Boolesche Formel in konjunktiver Normalform mit Variablen $x_1, \ldots, x_l$ und Klauseln $c_1, \ldots, c_k$, wobei jede Klausel genau drei Literale enthält. In der Codierung $\langle\phi\rangle$ einer Formeln ϕ seien die Variablen durch Zahlen in Binärdarstellung codiert. Dann sei $f(\langle\phi\rangle) = \langle S, t\rangle$ mit $S \subset \mathbb{N}$ und $t \in \mathbb{N}$, so dass folgende Bedingungen erfüllt sind:

1. Für jede Variable x_i von ϕ enthält S zwei Zahlen y_i und z_i und für jede Klausel c_j von ϕ enthält S zwei Zahlen g_j und h_j.
2. Die Dezimaldarstellung der Zahl y_i besitzt die Form $10^{l-i}p_{i1}p_{i2}\ldots p_{ik}$, wobei $p_{ij} = 1$ ist, falls die Variable x_i in der Klausel c_j positiv vorkommt, und $p_{ij} = 0$ sonst.
3. Die Dezimaldarstellung der Zahl z_i besitzt die Form $10^{l-i}n_{i1}n_{i2}\ldots n_{ik}$, wobei $n_{ij} = 1$ ist, falls die Variable x_i in der Klausel c_j negativ vorkommt, und $n_{ij} = 0$ sonst.
4. Die Dezimaldarstellung der Variablen g_j und h_j besitzt die Form 10^{k-j}.
5. Die Dezimaldarstellung von t besitzt die Form 1^l3^k.

(ii) Für jedes Wort w, dass keine Boolesche Formel in konjunktiver Normalform codiert, bei der jede Klausel genau drei Literal enthält, sei $f(w) = \langle S, t\rangle$ für ein beliebig gewähltes $\langle S, t\rangle \notin$ *SUBSET-SUM.*

Beweisen Sie, dass f eine Polynomzeit-Reduktion des Problems *3SAT* auf *SUBSET-SUM* ist.

Orakel-Turingmaschinen

Man kann sich die Frage stellen, wie sich die Berechnungsstärke oder die Komplexitätseigenschaften von Turingmaschinen unter der Annahme verändern würden, dass man ein gegebenes unentscheidbares Problem entscheiden *könnte*, oder dass ein Problem, von dem kein polynomieller Algorithmus bekannt ist, in polynomieller Zeit lösbar *wäre*. Es macht allerdings keinen Sinn, derartige Annahmen direkt zu treffen, weil sich dadurch eine widersprüchliche Annahmenmenge ergeben kann, aus der sich dann natürlich jede Schlussfolgerung ableiten lässt. Um diese Fragen trotzdem untersuchen zu können, verwendet man das Modell der Orakel-Turingmaschine:

Definition 1. Sei $A \subseteq \Sigma^*$ eine beliebige Sprache. Eine *Turingmaschine mit Orakel A* ist eine Turingmaschine mit drei speziellen Zuständen $q_?$, q_{ja} und q_{nein}. Der Zustand $q_?$ kann verwendet werden, um das Orakel zu „fragen", ob ein Wort auf dem Band (das längste Wort, das an der Kopfposition beginnt und nur aus Zeichen aus Σ besteht) in A enthalten ist. Das Orakel „antwortet", indem die Turingmaschine in einem Schritt in den Zustand q_{ja} übergeht, wenn das Wort in A enthalten ist, und in den Zustand q_{nein}, wenn das Wort nicht in A enthalten ist.

Eine Turingmaschine mit Orakel A kann also während der Berechnung jederzeit prüfen, ob ein Wort in der Sprache A enthalten ist. Diese Prüfung erfordert immer nur einen Berechnungsschritt. Dabei spielt weder eine Rolle, in welcher Komplexitätsklasse A liegt, noch ob A überhaupt entscheidbar oder Turing-akzeptierbar ist. Man bezeichnet eine Turingmaschine M mit Orakel A mit M^A. Die Menge der Sprachen, die in polynomieller Zeit mit einer deterministischen Turingmaschine mit Orakel A entschieden werden können, bezeichnet man mit P^A.

Hausaufgabe 11.3

Zeigen Sie, dass NP und coNP Teilmengen von $\mathrm{P}^{\mathrm{SAT}}$ sind und dass aus $\mathrm{NP} = \mathrm{P}^{\mathrm{SAT}}$ folgen würde, dass NP = coNP gilt.

Hausaufgabe 11.4

Beweisen Sie, dass P genau dann unter nichtlöschenden Homomorphismen abgeschlossen ist, wenn P = NP gilt.

Hinweis: Betrachten Sie folgende Sprache, die offensichtlich in P *liegt:* $S = \{xy \mid x \in \{0,1\}^*$ *ist die Codierung einer Booleschen Formel* ϕ *und* $y \in \{\mathrm{t}, \mathrm{f}\}^*$ *ist die Codierung einer Variablenbelegung, welche die Formel* ϕ *erfüllt*$\}$.

Hausaufgabe 11.5

Sei *UNARY-SUBSET-SUM* eine Variante des Teilmengensummen-Problems, bei dem alle Zahlen unär codiert sind. Wenn $\langle x_1, \ldots, x_n \rangle_u = 0^{x_1}1 \ldots 10^{x_n}$ die unäre Codierung der Zahlenfolge $x_1, \ldots, x_n$ ist, dann gilt also: $\langle S, t \rangle_u \in$ *UNARY-SUBSET-SUM* genau dann, wenn $\langle S, t \rangle \in$ *SUBSET-SUM*.

1. Warum lässt sich die in Aufgabe 11.2 konstruierte Polynomzeit-Reduktion von *3SAT* auf *SUBSET-SUM* nicht auf *UNARY-SUBSET-SUM* übertragen?

2. Beweisen Sie, dass *UNARY-SUBSET-SUM* in P liegt.