ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Prof. Dr. Andreas Podelski
Matthias Heizmann
Alexander Nutz
Christian Schilling

06.02.2014
Abgabe: Dienstag 11.02.2014, 16 Uhr
zu Beginn der Vorlesung

14. Übungsblatt zur Vorlesung
Theoretische Informatik

Aufgabe 1: Entscheidbarkeit 3 Punkte
Beweisen Sie die folgende Behauptung:
Wenn L eine rekursiv aufzählbare Sprache ist, die sich auf ihr Komplement reduzieren lässt (also $L \leq \overline{L}$ gilt), dann L ist entscheidbar.

Aufgabe 2: Entscheidbarkeit, Semi-Entscheidbarkeit 3+3 Punkte
Sei $B = \{0, 1\}$. Welche der folgenden Sprachen sind entscheidbar, welche sind rekursiv aufzählbar? Beweisen Sie Ihre Behauptungen.

(a) Hält die Turingmaschine τ auf mindestens einer Eingabe nicht?

$$\{bw_\tau \in B^* \mid \text{Es gibt ein } u \in B^*, \text{ sodass } \tau \text{ angesetzt auf } u \text{ nicht hält}\}$$

(b) Hält die Turingmaschine $\tau_{n,k}$ mit n Zuständen und k Bandsymbolen auf dem leeren Band?

$$\{bw_{\tau_{n,k}} \in B^* \mid \tau_{n,k} = (Q, B, \Gamma, \delta, q_0, \sqcup) \text{ mit } |Q| = n \text{ und } |\Gamma| = k \\ \text{angesetzt auf das leere Wort hält}\}$$

Aufgabe 3: Reduktion 3 Punkte
Welche der folgenden Reduktionen gelten? Beweisen Sie Ihre Behauptungen.

(a) $SAT \leq H$

(b) $H \leq SAT$

(c) $SAT \leq_p H$

(d) $H \leq_p SAT$

Dabei ist H das allgemeine Halteproblem für Turingmaschinen und SAT das Erfüllbarkeitsproblem für Boolesche Ausdrücke in konjunktiver Normalform.

Aufgabe 4: SAT $\in$ P ? 2 Punkte

Während der Erstellung der Aufgaben ist uns folgender Algorithmus eingefallen, der SAT in quadratischer Zeit löst und damit zeigt, dass P = NP ist. Gegeben sei eine boolesche Formel F mit den Variablen $x_1, \ldots, x_k$. Offensichtlich ist $k \leq |F|$.

1. Wir reduzieren das Problem, ob F erfüllbar ist, auf ein einfacheres Problem mit der Formel $F^{(1)}$, die $k-1$ Variablen enthält. Die Reduktion ersetzt jedes Vorkommen von x_k in F einmal durch 0 und einmal durch 1:

   $$F^{(1)} = F[x_k := 0] \vee F[x_k := 1]$$

   Der Algorithmus lässt sich auf einer Zweibandturingmaschine in $3 \cdot |F| + c$ Schritten durchführen, wobei c eine kleine Konstante ist. Insgesamt ist die Reduktion in $O(n)$. Außerdem ist $F^{(1)}$ genau dann erfüllbar, wenn F erfüllbar ist.

2. Wir wiederholen den ersten Schritt k mal, bis wir eine Formel $F^{(k)}$ erhalten, die keine Variablen mehr enthält. Der Algorithmus hat Laufzeit $\leq n \cdot O(n)$, also $O(n^2)$.

3. Im letzten Schritt berechnen wir den Wahrheitswert von $F^{(k)}$. Weil die Formel keine Variablen mehr enthält, ist das in linearer Zeit möglich. Der gesamte Algorithmus hat also Zeitkomplexität $O(n^2 + n) = O(n^2)$.

Wo liegt der Fehler?

Aufgabe 5: Polynomielle Reduktion 4 Punkte

Das NP-vollständige Problem HSet (Hitting-Set) ist wie folgt definiert:

> *Gegeben:* Eine Menge M und eine Menge von Teilmengen $\mathcal{S}$ (d.h. $\mathcal{S} = \{S_1, \ldots, S_n\}$, $S_i \subseteq M$ für $i = 1, \ldots, n$) sowie eine natürliche Zahl $k \leq n$.
>
> *Frage:* Gibt es eine Teilmenge $T \subseteq M$, sodass $|T| \leq k$ und $T \cap S_i \neq \varnothing$ für $i = 1, \ldots n$? (Mit anderen Worten: Gibt es eine höchstens k-elementige Teilmenge T, die mit jedem S_i mindestens ein gemeinsames Element hat?)

Das NP-vollständige Problem Knüb (Knotenüberdeckung, bzw. VertexCover) ist wie folgt definiert.

> *Gegeben:* Ein ungerichteter Graph $G = (V, E)$ und eine natürliche Zahl k.
>
> *Frage:* Besitzt G eine "überdeckende Knotenmenge" der Größe höchstens k? (Eine überdeckende Knotenmenge ist eine Teilmenge $V' \subseteq V$, sodass für alle Kanten $(u, v) \in E$ gilt: $u \in V'$ oder $v \in V'$.)

(a) Begründen Sie, warum HSet in NP liegt.

(b) Beweisen Sie, dass HSet NP-vollständig ist. Sie dürfen dabei verwenden, dass Knüb NP-vollständig ist.

Es genügt, wenn Sie Ihre Laufzeitabschätzung grob begründen. Sie müssen weder Pseudocode noch Turingmaschinen explizit angeben.