Goethe-Universität Frankfurt am Main
Institut für Informatik
Theorie komplexer Systeme
Prof. Dr. Nicole Schweikardt

Dienstag, 03.05.2011

Komplexitätstheorie

Sommersemester 2011

Übungsblatt 3

Zu bearbeiten bis Donnerstag, 12.05.2011

Aufgabe 1: **(25 Punkte)**

Ist C eine der beiden Komplexitätsklassen EXP und NEXP, so heißt ein Problem L' C–*hart*, falls $L \leqslant_p L'$ für jedes $L \in$ C gilt. L' heißt C-*vollständig*, falls $L' \in$ C und L' C–hart ist.

(a) Zeigen Sie, dass die in Kapitel 3.3 betrachtete Sprache

$$\textbf{EXPCOM} := \{\langle \alpha, x, 1^t \rangle : \alpha, x \in \{0,1\}^*, t \in \mathbb{N}, \text{ so dass } M_\alpha \text{ bei Eingabe } x \text{ nach} \leqslant 2^t \text{ Schritten 1 ausgibt}\}$$

EXP–vollständig ist.

(b) Geben Sie eine Sprache L' an, die NEXP–vollständig ist und weisen Sie deren NEXP–Vollständigkeit nach.

(c) Zeigen Sie: Falls eine NEXP–vollständige Sprache L' in EXP liegt, so ist EXP = NEXP.

Aufgabe 2: **(25 Punkte)**

Beweisen Sie die Behauptung 0 aus dem Beweis des Satzes von Ladner (Theorem 3.6 aus der Vorlesung). Das heißt, zeigen Sie: Es gibt eine deterministische Polynomialzeit–Turingmaschine, die die Funktion H berechnet.

Aufgabe 3: **(25 Punkte)**

Zeigen Sie, dass die folgende Sprache L unentscheidbar ist:

$$L := \{\alpha \in \{0,1\}^* : M_\alpha \text{ hält nach höchstens } 100n^2 + 200 \text{ Schritten}\}.$$

([AB], Aufgabe 3.1)

— auf der nächsten Seite geht's weiter —

Aufgabe 4: **(25 Punkte)**

Sei φ eine aussagenlogische Formel, in der genau die aussagenlogischen Variablen aus $\{x_1, \ldots, x_n\}$ vorkommen. Für j mit $0 \leqslant j \leqslant n$ und $z := z_1 \cdots z_j \in \{0,1\}^j$ schreiben wir φ_z, um die aussagenlogische Formel zu bezeichnen, die aus φ entsteht, indem jede Variable x_i (für $1 \leqslant i \leqslant j$) durch den Wert $z_i \in \{0,1\}$ ersetzt wird. Es ist klar, dass φ genau dann erfüllbar ist, wenn ein $z \in \{0,1\}^n$ existiert, so dass $\varphi_z \equiv 1$.

Sei $h_\varphi : \{0,1\}^* \to \{1\}^*$ eine Funktion mit der Eigenschaft, dass für alle $z, z' \in \{0,1\}^*$ mit $|z|, |z'| \leqslant n$ aus $h_\varphi(z) = h_\varphi(z')$ folgt, dass φ_z genau dann erfüllbar ist, wenn $\varphi_{z'}$ erfüllbar ist.

Für eine Folge T von Paaren definieren wir die folgenden Funktionen: $lookup(T, k)$ sei v, falls (k, v) in der Folge T vorkommt, und **false** sonst; $contains(T, k)$ sei **false**, falls $lookup(T, k) =$ **false**, und **true** sonst; falls $T = P(k, v')S$, für Folgen P und S, so sei $insert(T, k, v) := P(k, v)S$, ansonsten sei $insert(T, k, v) := T(k, v)$.

In den folgenden Algorithmen ist T eine „globale Variable“, deren Belegung innerhalb der rekursiven Aufrufe verändert werden kann.

Algorithmus 1 $isSat(\varphi)$

```
1: global T ← ()
2: z ← ϵ
3: return isSat'(φ, z).
```

Algorithmus 2 $isSat'(\varphi, z)$

```
1: if |z| = n then
2:     sat ← (φ_z ≡ 1)
3: else if contains(T, h_φ(z)) then
4:     sat ← lookup(T, h_φ(z))
5: else if isSat'(φ, z0) or isSat'(φ, z1) then
6:     sat ← true
7: else
8:     sat ← false
9: end if
10: T ← insert(T, (h_φ(z), sat))
11: return sat
```

(a) Zeigen Sie, dass für jede aussagenlogische Formel φ, in der genau die Variablen aus $\{x_1, \ldots, x_n\}$ vorkommen, gilt:

   (i) $isSat(\varphi) =$ **true** $\iff$ φ ist erfüllbar.

   (ii) Wenn die Funktion h_φ in Polynomialzeit berechenbar ist, so terminiert $isSat(\varphi)$ nach poly$(|\varphi|)$ vielen Schritten.

(b) Zeigen Sie den folgenden Satz von Berman (1978):
Wenn es eine NP–vollständige Sprache $L \subseteq \{1\}^*$ gibt, dann ist P = NP.

Beachten Sie hierbei: Falls $L = \emptyset$ oder $L = \{1\}^*$ so ist offensichtlicherweise $L \in P$ und aus der NP–Vollständigkeit von L folgt P = NP. Zeigen Sie zunächst: Falls $\emptyset \subsetneq L \subsetneq \{1\}^*$, so gibt es für jedes $L' \in$ NP eine Polynomialzeit–Reduktion f von L' auf L, die jedes Wort in $\{0,1\}^*$ auf ein Wort $f(x) \in \{1\}^*$ abbildet.