AG Theorie der künstlichen Intelligenz
FB Mathematik und Informatik, Universität Bremen
Prof. Dr. Carsten Lutz

MZH 3090
clu@uni-bremen.de
Tel.: 0421/218-64431

2. Aufgabenblatt für die Vorlesung „Komplexitätstheorie“

Aufgabe 1: 25%

Ein nicht-deterministischer endlicher Automat (NEA) heißt *azyklisch (ANEA)* wenn seine Übergangsrelation azyklisch ist, d.h. es gibt keine Folge $a_0, \dots, a_{n-1}$ von Symbolen und $q_0, \dots, q_n$ von Zuständen so dass $q_0 = q_n$ und

$$(q_1, a_i, q_{i+1}) \in \Delta \text{ für alle } i < n.$$

Beweise durch Angeben eines polynomiellen Beweissystems, dass das folgende Problem in NP ist: gegeben ANEAs $\mathcal{A}_1$ und $\mathcal{A}_2$, gilt $L(\mathcal{A}_1) \not\subseteq L(\mathcal{A}_2)$?

Aufgabe 2: 24%

Beweise die folgenden Aussagen aus der Vorlesung:

(a) $B \in \mathrm{P}$ und $A \leq_p B$ impliziert $A \in \mathrm{P}$;

(b) $B \in \mathrm{NP}$ und $A \leq_p B$ impliziert $A \in \mathrm{NP}$;

(c) $A \leq_p B$ und $B \leq_p C$ impliziert $A \leq_p C$.

Aufgabe 3: 26%

Für eine Sprache L ist $L_0 = \{\varepsilon\}$, $L_{i+1} = L_i \cdot L$ für alle $i \geq 0$ und $L^* = \bigcup_{i \geq 0} L_i$. Beweise:

(a) Wenn $L \in \mathsf{P}$, dann $L^* \in \mathsf{P}$;

(b) Wenn $L \in \mathsf{NP}$, dann $L^* \in \mathsf{NP}$.

Es genügt, Algorithmen in Pseudocode (und nicht als TM) anzugeben.

Hinweis: für (a) hilft dynamische Programmierung.

Aufgabe 4: 25%

Beweise durch Reduktion von 3SAT, dass IPROG NP-hart ist (beachte die Hinweise auf den Folien).

Zur Erinnerung: IPROG läßt nur Gleichungen der Form $c_1 \cdot x_1 + \cdots + c_n \cdot x_n = \alpha$ zu, wobei $x_1, \dots, x_n \in V$, $c_1, \dots, c_n \in \mathbb{N}$ und $\alpha \in V \cup \mathbb{N}$. Möchte man z.B. auch “$\leq$” und “$\geq$” verwenden, so muß man dies vorher als Abkürzung definieren.

Aufgabe 5: 25% (Zusatzaufgabe)

Sei φ eine AL-Formel. Eine $\neq$-*Wertzuweisung* für die Variablen in φ ist eine Wertzuweisung so dass jede Klausel in φ mindestens zwei Literale mit ungleichen Wahrheitswerten enthält. Mit anderen Worten: eine $\neq$-Wertzuweisung erfüllt φ ohne in irgendeiner Klausel alle drei Literale wahr zu machen. Sei $\neq$3SAT die Menge aller 3-Formeln, für die es eine $\neq$-WZ gibt. Beweise, dass $\neq$3SAT NP-vollständig ist.

Hinweis: Für die NP-Härte verwende eine Reduktion von 3SAT, die jede 3SAT-Klausel in zwei $\neq$3SAT-Klauseln umwandelt. Um die Korrektheit der Reduktion zu beweisen, hilft es wahrscheinlich, folgende Beobachtung auszunutzen: wenn man die Wahrheitswerte einer $\neq$-WZ für eine AL-Formel φ vertauscht (wahr durch falsch, falsch durch wahr), erhält man wieder eine $\neq$-WZ für φ.