

Theoretische Informatik (WS 2005/2006) - Übungsblatt 9

Jan Hendrik Dithmar - Matrikelnummer 2031259
Do, 11-13 Uhr, Vasileios Anastasatos

Aufgabe 1

Es sei $\Sigma = \{0, 1, \$\}$. Wir betrachten

$$L = \{u\$v \mid u \in SAM \text{ und } v \in \overline{SAM}\}$$

Weder L noch $\overline{L}$ sind akzeptierbar.

- Zu zeigen: $\overline{SAM} \preceq L$.

Dazu konstruieren wir eine TM-berechenbare Funktion $f : \Sigma^* \rightarrow \Sigma^*$ mit $x \in \overline{SAM} \Leftrightarrow f(x) \in L$.

Es sei nun M_1 Turing-Maschine mit $M_1 \in SAM$.

Wir setzen $f(x) = \langle M_1 \rangle \x . Dann gilt:

$$\begin{aligned} x \in \overline{SAM} &\Rightarrow f(x) = \langle M_1 \rangle \$x \in L \\ x \notin \overline{SAM} &\Rightarrow x \in SAM \Rightarrow f(x) = \langle M_1 \rangle \$x \notin L \end{aligned}$$

Aus der Vorlesung wissen wir, dass $\overline{SAM}$ nicht akzeptierbar ist. Mit der Behauptung

$$L_1 \preceq L_2 \text{ und } L_1 \text{ nicht akzeptierbar} \Rightarrow L_2 \text{ nicht akzeptierbar}$$

sehen wir, dass L nicht akzeptierbar ist.

- Zu zeigen: $\overline{SAM} \preceq \overline{L}$.

Dazu konstruieren wir eine Turing-Maschine $M_2 \notin SAM$. Wir setzen $f(x) = x\$ \langle M_2 \rangle$. Dann gilt:

$$\begin{aligned} x \in \overline{SAM} &\Rightarrow f(x) = x\$ \langle M_2 \rangle \notin L \Rightarrow f(x) \in \overline{L} \\ x \notin \overline{SAM} &\Rightarrow x \in SAM \Rightarrow f(x) = x\$ \langle M_2 \rangle \in L \Rightarrow f(x) \notin \overline{L} \end{aligned}$$

Wie oben folgt, dass $\overline{L}$ nicht akzeptierbar ist.

Zeige, dass gilt:

- $L_1 \preceq L_2$ und L_2 akzeptierbar $\Rightarrow L_1$ akzeptierbar
 L_2 sei akzeptierbar, d.h. es gibt eine Turing-Maschine M_2 , die L_2 akzeptiert. $L_1 \preceq L_2$ bedeutet, dass es eine berechenbare Funktion f gibt mit $x \in L_1 \Leftrightarrow f(x) \in L_2$. Diese werde realisiert durch die Turing-Maschine M_1 . Wir konstruieren daraus eine Turing-Maschine M , die L_1 akzeptiert. Es sei eine Eingabe x gegeben. Zuerst simuliert M die Maschine M_1 und berechnet $f(x)$ (M_1 terminiert für alle $x \in L_1$), danach simuliert sie M_2 bei Eingabe $f(x)$. Sie akzeptiert $f(x)$ genau dann, wenn $f(x) \in L_2$ ist, was gleichbedeutend mit $x \in L_1$ ist. Also akzeptiert M eine Eingabe x genau dann, wenn $x \in L_1$.
- $L_1 \preceq L_2$ und L_1 nicht akzeptierbar $\Rightarrow L_2$ nicht akzeptierbar
Nehmen wir an, L_2 wäre akzeptierbar. Dann folgt aus obiger Implikation, dass L_1 akzeptierbar wäre, was aber der Voraussetzung widerspricht.

Aufgabe 2

(a)

Eine BB-TM kann die Sprache $\overline{\text{UNIV}}$ akzeptieren, indem das Ergebnis invertiert wird, d.h., sagt die BB-TM für UNIV ja (ist also in Zustand q_{ja}), dann sagt die BB-TM für $\overline{\text{UNIV}}$ nein (ist also im Zustand q_{nein}) und umgekehrt.

(b)

Aufgabe 3

(a)

Idee: Das Problem ist entscheidbar.

Sei (x_i, y_i) für $i = 1, \dots, n$ eine Instanz des PCP_Σ . Die Frage, ob es eine Indexfolge $(i_1, \dots, i_k)$ mit $k \in \mathbb{N}, i_1, \dots, i_k \in \{1, \dots, n\}$ mit $x_{i_1}, \dots, x_{i_k} = y_{i_1}, \dots, y_{i_k}$ gibt, ist äquivalent zu der Frage, ob es eine Indexfolge $(i_1, \dots, i_k)$ gibt mit

$$\begin{aligned} |x_{i_1} \dots x_{i_k}| &= |y_{i_1} \dots y_{i_k}| \\ &= |x_{i_1}| + \dots + |x_{i_k}| \\ &= |y_{i_1}| + \dots + |y_{i_k}| \end{aligned}$$

Dann gilt:

$$\text{PCP}_\Sigma \text{ unlösbar} \Leftrightarrow \forall i : |x_i| < |y_i| \text{ oder } \forall i : |y_i| < |x_i|$$

• $\Rightarrow$:

Sei PCP_Σ unlösbar. Falls es ein (x_i, x_j) mit $|x_i| = |x_j|$ gibt, dann besitzt PCP_Σ eine Lösung (Widerspruch zur Annahme). Es gelte deshalb nun $|x_i| \neq |y_i| \forall i$. Wir nehmen an, dass es ein j und ein k gibt mit $|x_j| > |y_j|, |x_k| < |y_k|$. Betrachte dazu

$$\begin{aligned} n_1 &:= |x_j| - |y_j| \\ n_2 &:= |y_k| - |x_k| \end{aligned}$$

Es gilt nun:

$$\begin{aligned} |x_j^{n_2} x_k^{n_1}| &= (n_1 + |y_j|) \cdot n_2 + (|y_k| - n_2) \cdot n_1 \\ &= n_2 \cdot |y_j| + n_1 \cdot |y_k| \\ &= |y_j^{n_2} y_k^{n_1}| \end{aligned}$$

Es gilt, dass $j^{n_2} \cdot k^{n_1}$ eine Lösung des PCP_Σ ist.

- $\Leftarrow$:

Sei nun $|x_i| < |y_i| \forall i$. Dann gilt $\forall (i_1, \dots, i_k), k \geq 1$:

$$\begin{aligned} |x_{i_1} \dots x_{i_k}| &= |x_{i_1}| + \dots + |x_{i_k}| \\ &< |y_{i_1}| + \dots + |y_{i_k}| \\ &= |y_{i_1} \dots y_{i_k}| \end{aligned}$$

Also kann es keine nichttriviale Lösung geben.

$\Rightarrow$ Um PCP_Σ zu entscheiden, genügt es $|y_i| < |x_i|$ für alle i zu testen.

(b)

Idee: Das Problem ist entscheidbar.

Sei $|\Sigma| = 3$; $0 < |x_i| \leq 5$.

Auf einer Kartenhälfte können nur $\underbrace{3}_{|x_i|=1} + \underbrace{3^2}_{|x_i|=2} + \underbrace{3^3}_{|x_i|=3} + \underbrace{3^4}_{|x_i|=4} + \underbrace{3^5}_{|x_i|=5} = 363$ ver-

schiedene Strings stehen. Es gibt nur 363^2 verschiedene Karten. Es gilt: $k \leq 363^2$, denn es kann nicht mehr verschiedene Karten geben, als es überhaupt Karten geben kann. Für ein k gibt es nur endlich viele Instanzen des PCP mit k Spielkarten, genauer $\binom{363^2}{k}$. Die Summe aller Instanzen ist $\sum_{k=0}^{363^2} \binom{363^2}{k}$, was wiederum endlich ist. Die Gesamtzahl aller Instanzen ist gerade $2^{(363^2)}$, dies ist nämlich die Mächtigkeit der Potenzmenge einer 363^2 -elementigen Menge. Um eine endliche Menge zu entscheiden, genügt ein endlicher Automat. Ein endlicher Automat kann von einer Turing-Maschine simuliert werden. Reguläre Sprachen sind unter Komplementbildung abgeschlossen. Das Problem ist also entscheidbar.