7. Aufgabenblatt

(Besprechung in den Tutorien 05.12.2022–09.12.2022)

Aufgabe 1. Das AKT-Problem

Sei M_w die Turing-Maschine, die durch das Wort $w \in \{0,1\}^*$ kodiert wird. Zeigen oder widerlegen Sie, dass die Sprache

$$L_{\mathrm{AKT}} = \{w \in \{0,1\}^* \mid M_w \text{ hält auf Eingabe } AKT\}$$

entscheidbar ist.

——————Lösungsskizze——————

Wir reduzieren das Halteproblem auf leerem Band $H_0 = \{w \in \{0,1\}^* \mid M_w$ hält bei leerer Eingabe$\}$ auf die Sprache L_{AKT} (wir zeigen also $H_0 \leq L_{\mathrm{AKT}}$). Daraus folgt, dass L_{AKT} unentscheidbar ist.

Für eine Turing-Maschine $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ definieren wir eine Turing-Maschine $\lambda(M)$, die auf einer Eingabe x wie folgt arbeitet:

1. Lösche die Eingabe x vom Band.
2. Simuliere das Verhalten von M.

Formal ist $\lambda(M) = (Z \cup \{z_s\}, \Sigma, \Gamma, \delta', z_s, \square, E)$, wobei $z_s \notin Z$ ein neuer Startzustand ist und δ' wie folgt definiert ist:

- Für alle $z \in Z$ und $a \in \Gamma$ ist $\delta'(z, a) := \delta(z, a)$,
- für alle $a \in \Sigma$ ist $\delta'(z_s, a) := (z_s, \square, R)$ und
- $\delta'(z_s, \square) := (z_0, \square, N)$.

Nun definieren wir die Reduktionsfunktion $f\colon \{0,1\}^* \to \{0,1\}^*$ wie folgt: $f(w) := w'$, wobei $w' := \langle \lambda(M_w) \rangle$ ($M_{w'}$ ist also die Turing-Maschine $\lambda(M_w)$).

Wir beobachten: f ist total, da nach Definition der Kodierung $\langle \cdot \rangle$ von Turing-Maschinen für jedes w eine Turing-Maschine M_w existiert und somit auch $M_{w'}$ und ihre Kodierung w'. Weiter ist f berechenbar, wir müssen nur die Kodierung der Turing-Maschine $\lambda(M_w)$ zurückgeben.

Um zu zeigen, dass f eine Reduktion von H_0 auf L_{AKT} ist, müssen wir zeigen, dass

$$\forall w \in \{0,1\}^* : w \in H_0 \iff f(w) \in L_{\mathrm{AKT}}.$$

Zunächst zeigen wir: $w \in H_0 \Rightarrow f(w) \in L_{\mathrm{AKT}}$. Wenn $w \in H_0$ ist, dann hält M_w auf leerer Eingabe. Dann hält per Konstruktion auch $M_{w'}$ auf Eingabe AKT, also ist $f(w) \in L_{\mathrm{AKT}}$.

Nun zeigen wir: $w \notin H_0 \Rightarrow f(w) \notin L_{\mathrm{AKT}}$. Sei also $w \notin H_0$. Dann hält M_w auf leerer Eingabe nicht und somit hält auch $M_{w'}$ nicht auf Eingabe AKT, also ist $f(w) \notin L_{\mathrm{AKT}}$.

Aufgabe 2. Totalitätsproblem

Sei M_w die Turing-Maschine, die durch das Wort $w \in \{0,1\}^*$ kodiert wird. Zeigen Sie durch eine Reduktion vom speziellen Halteproblem $K = \{w \in \{0,1\}^* \mid M_w \text{ hält auf Eingabe } w\}$, dass die Sprache

$$T = \{w \in \{0,1\}^* \mid M_w \text{ hält bei jeder möglichen Eingabe } x \in \{0,1\}^*\}$$

unentscheidbar ist. Diskutieren Sie, was aus dieser Erkenntnis folgt.

——————Lösungsskizze——————

Wir zeigen $K \leq T$.

Sei M eine beliebige Turing-Maschine. Wir konstruieren eine neue Turing-Maschine $\alpha(M)$, die genau dann bei *jeder möglichen* Eingabe hält, wenn M bei Eingabe $\langle M \rangle$ hält. Die Turing-Maschine $\alpha(M)$ geht dabei wie folgt vor:

1. Lösche die Eingabe vom Band (mittels neuem Startzustand).
2. Schreibe das Wort $\langle M \rangle$ auf das Band (mit $|\langle M \rangle|$ neuen Zuständen) und bewege den Lese-/Schreibkopf auf das erste Symbol von $\langle M \rangle$ (ein weiterer Zustand, der "zurückspult" und in den Startzustand von M überführt).
3. Simuliere das Verhalten von Turing-Maschine M.

Wir definieren die Reduktionsfunktion $f\colon \{0,1\}^* \to \{0,1\}^*$, $w \mapsto \langle \alpha(M_w) \rangle$ und beobachten, dass diese total und berechenbar ist (Kodieren von TM). Es bleibt zu zeigen, dass

$$\forall w \in \{0,1\}^* : w \in K \iff f(w) \in T.$$

Wenn nun $w \in K$ ist, also M_w bei Eingabe w nach endlich vielen Schritten hält, dann hält $\alpha(M_w)$ nach endlich vielen Schritten bei jeder beliebigen Eingabe; also ist $f(w) \in T$. Wenn hingegen $w \notin K$ ist, also M_w bei Eingabe w in eine Endlosschleife gerät, dann gerät auch $\alpha(M_w)$ bei jeder möglichen Eingabe in eine Endlosschleife; also ist $f(w) \notin T$.

Aus der Unentscheidbarkeit von T folgt: Wir können keinen Compiler bauen, der entscheidet, ob ein Programm in eine Endlosschleife geraten könnte.

Aufgabe 3. Besuchen von Zuständen

Eine Turing-Maschine *besucht* einen Zustand z bei einer Eingabe w, falls die dazugehörige Konfigurationsfolge eine Konfiguration enthält, die z enthält.

Zeigen oder widerlegen Sie, dass folgende Sprache entscheidbar ist:

$$L = \left\{ w\#x\#y \;\middle|\; \begin{array}{l} w,x,y \in \{0,1\}^* \text{ und } y \text{ ist die Kodierung eines Nicht-Endzustandes } z \\ \text{von } M_w \text{ und } M_w \text{ besucht } z \text{ bei Eingabe } x \end{array} \right\}.$$

——————Lösungsskizze——————

Wir reduzieren vom allgemeinen Halteproblem $H = \{w\#x \mid M_w \text{ hält auf Eingabe } x \in \{0,1\}^*\}$. Wir zeigen also $H \leq L$, woraus folgt, dass L unentscheidbar ist.

Sei $M = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$ eine Turing-Maschine.

Intuitiv konstruieren wir nun eine Turing-Maschine $\alpha(M)$, deren Unterschied zu M ist, dass sie (1) einen neuen Zustand z' hat, (2) keinen Endzustand hat und (3) falls M halten würde, in den Zustand z' übergeht.

Wir definieren die Turing-Maschine $\alpha(M) = (Z \cup \{z'\}, \Sigma, \Gamma, \delta', z_0, \square, \emptyset)$, wobei $z' \notin Z$ und

$$\delta'(z,\gamma) := \begin{cases} \delta(z,\gamma), & \text{falls } \delta(z,\gamma) \neq \bot \\ (z',\gamma,N), & \text{sonst} \end{cases}$$

für $z \in Z$ und $\gamma \in \Gamma$.

Beobachte, dass $\alpha(M')$ für jede beliebige Turing-Maschine M' berechnet werden kann, ohne M' zu simulieren, da wir lediglich einen Zustand hinzufügen, die Endzustände in Nicht-Endzustände umwandeln und die Übergangsfunktion entsprechend anpassen.

Nun sei $f\colon \{0,1,\#\}^* \to \{0,1,\#\}^*$ wie folgt definiert: Sei $s \in \{0,1,\#\}^*$. Falls $s = w\#x$ und $w,x \in \{0,1\}^*$ ist, dann ist $f(s) = \langle \alpha(M_w) \rangle \#x\#y$ wobei y die Kodierung des neuen Zustands z' in $\alpha(M_w)$ ist. Andernfalls, ist $f(s) = 0$.

Wir stellen fest, dass f total und berechenbar ist. Es bleibt zu zeigen, dass

$$\forall s \in \{0,1,\#\}^* : s \in H \iff f(s) \in L.$$

Falls $s \neq w\#x$ für $w, x \in \{0,1\}^*$, dann gilt offensichtlich $s \notin H$ und $f(s) = 0 \notin L$. Andernfalls ($s = w\#x$), gilt per Konstruktion

$$s \in H \implies M_w \text{ hält auf } x \implies \alpha(M_w) \text{ besucht } z' \implies f(s) \in L$$

und

$$s \notin H \implies M_w \text{ hält nicht auf } x \implies \alpha(M_w) \text{ besucht nie } z' \implies f(s) \notin L,$$

wobei z' der neue Zustand von $\alpha(M_w)$ ist.
