Universität Leipzig
Institut für Informatik
Sommersemester 2025

Prof. Dr. Andreas Maletti,
Dr. habil. Karin Quaas,
Fabian Sauer

Aufgaben zur Lehrveranstaltung

Berechenbarkeit

Lösungen zu Serie 6

Übungsaufgabe 6.1 (Entscheidbarkeit und Berechenbarkeit)

Beweisen oder widerlegen Sie die folgenden Aussagen:

(a) Die charakteristische Funktion χ_L jeder Menge $L \subseteq \Sigma^*$ ist Turing-berechenbar. Falsch. Die charakteristische Funktion des allgemeinen Halteproblems H ist nicht berechenbar, denn H ist nicht entscheidbar. (Ein Problem ist per Definition entscheidbar falls seine charakteristische Funktion berechenbar ist.)

(b) Das Komplement $\overline{H}$ des allgemeines Halteproblems ist nicht semientscheidbar. Wahr. Denn angenommen wäre $\overline{H}$ semientscheidbar. Das allgemeine Problem H ist semientscheidbar. Dann aber ist nach §8.8 H entscheidbar, Widerspruch zu §9.10 (H ist unentscheidbar).

(c) Jede Teilmenge einer rekursiv aufzählbaren Sprache ist abzählbar. Wahr. Jede rekursiv aufzählbare Sprache ist auch abzählbar. Jede Teilmenge eine abzählbaren Sprache ist abzählbar.

(d) Jede Teilmenge einer abzählbaren Sprache ist rekursiv aufzählbar. Falsch. Die Sprache $\{0,1\}^*$ ist abzählbar. Aber die Teilmenge $\overline{H}$ (Komplement des allgemeinen Halteproblems) ist nicht rekursiv aufzählbar.

Übungsaufgabe 6.2 (Entscheidbarkeit und Berechenbarkeit)

Sei $L \subseteq \{a,b\}^*$ rekusiv aufzählbar. Zeigen Sie, dass die folgenden beiden Sprachen rekursiv aufzählbar sind.

(a) $L_1 = \{w \cdot a \mid w \in L\}$

(b) $L_2 = \{u \cdot v \mid u, v \in L\}$

LÖSUNG: Da L rekursiv aufzählbar ist, gilt entweder $L = \emptyset$ (dann auch $L_1 = \emptyset$ und $L_2 = \emptyset$, also rekursiv aufzählbar und wir sind fertig) oder es existiert berechenbare surjektive Funktion $\mathbf{a} : \mathbb{N} \to L$.

(a) Definiere $\mathbf{a}_1 : \mathbb{N} \to L_1$ durch $n \mapsto \mathbf{a}(n) \cdot a$, für alle $n \in \mathbb{N}$. Wir beweisen, dass $\mathbf{a}_1$ berechenbar und surjektiv ist. (Berechenbarkeit:) Sei M eine deterministische Turingmaschine, welche $\mathbf{a}$ berechnet: bei Eingabe von n stoppt M also in q_+ mit

Bandinhalt $\mathbf{a}(n)$. Sei M' die deterministische Turingmaschine, welche bei Eingabe von n die Turingmaschine M mit Eingabe n simuliert; wenn M in q_+ mit $\mathbf{a}(n)$ stoppt, schreibt M' im Anschluss noch den Buchstaben a an das Ende von $\mathbf{a}(n)$ und geht dann in den akzeptierenden Zustand q'_+ mit Bandinhalt $\mathbf{a}(n) \cdot a$. M' berechnet also $\mathbf{a}_1$. (Surjektivität:) Sei nun $w \in L_1$. Also $w = u \cdot a$, wobei $u \in L$. Da $\mathbf{a}$ surjektiv, gibt es $n \in \mathbb{N}$ mit $\mathbf{a}(n) = u$. Also gilt auch $\mathbf{a}_1(n) = w$. Also ist $\mathbf{a}_1$ surjektiv.

(b) Wir benutzen die bijektive (und berechenbare) Cantorsche Paarfunktion $c : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ definiert durch $(n_1, n_2) \mapsto ((n_1 + n_2)(n_1 + n_2 + 1))/2 + 2$ für alle $n_1, n_2 \in \mathbb{N}$. Klarerweise ist auch die Umkehrfunktion c^{-1} bijektiv und berechenbar. Definiere $\mathbf{a}_2 : \mathbb{N} \to L_2$ durch $n \mapsto \mathbf{a}(\pi_1(c^{-1}(n))) \cdot \mathbf{a}(\pi_2(c^{-1}(n)))$ für alle $n \in \mathbb{N}$. Wir beweisen, dass $\mathbf{a}_2$ berechenbar und surjektiv ist: (Berechenbarkeit:) ... (Surjektivität:) Sei $w \in L_2$. Dann also existieren $u, v \in L$ mit $w = u \cdot v$. Da $\mathbf{a}$ surjektiv ist, existieren n_1 und n_2 mit $\mathbf{a}(n_1) = u$ und $\mathbf{a}(n_2) = v$. Sei $n = c(n_1, n_2)$. Dann also $\pi_1(c^{-1}(n)) = n_1$ und $\pi_2(c^{-1}(n)) = n_2$. Also $\mathbf{a}_2(n) = u \cdot v = w$. Also ist $\mathbf{a}_2$ surjektiv.

Übungsaufgabe 6.3 (Reduktion)

Definiere die Sprache $H_{\text{pal}} \subseteq \mathfrak{B}^*$ durch

$$H_{\text{pal}} = \{w \in \mathfrak{B}^* \mid \text{decode}(w) \text{ hält auf } w' \text{ falls } w' \text{ Palindrom ist}\}$$

Zeigen Sie durch Reduktion von H_ε (Halteproblem auf leerem Band) dass H_{pal} unentscheidbar ist. (Wiederholung: $H_\varepsilon = \{w \in \mathfrak{B}^* \mid \text{decode}(w) \text{ hält auf } \varepsilon\}$.) Um zu zeigen, dass H_{pal} unentscheidbar ist, genügt es nach §9.10 zu zeigen, dass H_ε auf H_{pal} reduzierbar ist, also $H_\varepsilon \preceq H_{\text{pal}}$. Wir definieren dafür eine totale berechenbare Funktion $f : \mathfrak{B}^* \to \mathfrak{B}^*$ sodass $w \in H_\varepsilon$ gdw. $f(w) \in H_{\text{pal}}$ für alle $w \in \mathfrak{B}^*$.

Für alle $w \in \mathfrak{B}^*$ definiere $f(w) = \text{code}(M_w)$, wobei M_w eine bereinigte deterministische Turingmaschine ist, welche bei Eingabe $v \in \mathfrak{B}^*$ zunächst v vom Band löscht, und dann die Turingmaschine $M := \text{decode}(w)$ auf leerem Band simuliert. Die Funktion f ist berechenbar.

Wir zeigen die Korrektheit der Reduktion, d.h., für alle $w \in \mathfrak{B}^*$ gilt $w \in H_\varepsilon$ gdw. $f(w) \in H_{\text{pal}}$. Sei also $w \in \mathfrak{B}^*$.

Angenommen $w \in H_\varepsilon$, d.h. $\text{decode}(w)$ hält auf leerem Band. Zu zeigen: $f(w) \in H_{\text{pal}}$, d.h. $M' := \text{decode}(f(w))$ hält bei Eingabe von v falls v ein Palindrom ist. Sei also $v \in \mathfrak{B}^*$ ein Palindrom. Dann löscht M' zunächst v vom Band und simuliert dann $M := \text{decode}(w)$ auf leerem Band. Nach Annahme hält also auch M'. Also gilt in der Tat $f(w) \in H_{\text{pal}}$. (Eigentlich haben wir etwas stärkeres gezeigt: M' hält bei Eingabe jeden Wortes!)

Angenommen $w \notin H_\varepsilon$, d.h. $\text{decode}(w)$ hält nicht auf leerem Band. Zu zeigen: $f(w) \notin H_{\text{pal}}$, d.h $M' := \text{decode}(f(w))$ hält bei Eingabe von mindestens einem Palindrom v nicht. Wir zeigen etwas stärkeres, nämlich, dass M' bei Eingabe gar keines Wortes hält: sei $v \in \mathfrak{B}^*$. M' löscht zunächst v vom Band und simuliert dann

$M := \mathrm{decode}(w)$ auf leerem Band. Nach Annahme hält M' nicht. Also gilt in der Tat $f(w) \notin H_{\mathrm{pal}}$.

Testfrage: $H_\forall = \{w \mid \mathrm{decode}(w)$ hält auf allen Wörtern$\}$ unentscheidbar?

Übungsaufgabe 6.4 (Satz von Rice)

Für die folgenden Sprachen L_i prüfen Sie bitte, ob der Satz von Rice anwendbar ist, um die Unentscheidbarkeit von L_i zu zeigen.

(a) $L_1 = \{w \in \mathfrak{B}^* \mid \mathrm{decode}(w)$ hält nicht bei Eingabe $0\}$ Um zu zeigen, dass der Satz von Rice anwendbar ist, müssen wir eine Menge $\mathcal{F}_1$ von partiellen berechenbaren Funktionen finden, sodass

- $\emptyset \subsetneq \mathcal{F}_1 \subsetneq \mathcal{R}$ (wobei $\mathcal{R}$ die Menge aller partiellen berechenbaren Funktionen ist), und
- $\mathcal{C}(\mathcal{F}_1) = L_1$ gilt ($\mathcal{C}(\mathcal{F}_1) = \{w \in \mathfrak{B}^* \mid T(\mathrm{decode}(w)) \in \mathcal{F}_1\}$ ist die Menge aller Kodierungen von deterministischen Turingmaschinen, die Funktionen in $\mathcal{F}_1$ berechnen, siehe Folie 22 in VL 9).

Definiere $\mathcal{F}_1 = \{f : \mathfrak{B}^* \dashrightarrow \mathfrak{B}^* \mid f$ berechenbar und $f(0)$ ist nicht definiert$\}$. Es gilt $\emptyset \subsetneq \mathcal{F}_1$, denn $\bot \in \mathcal{F}_1$. Es gilt $\mathcal{F}_1 \subsetneq \mathcal{R}$, denn $\mathrm{id} \in \mathcal{R} \setminus \mathcal{F}_1$. Weiterhin gilt

$$\begin{aligned}
& \mathcal{C}(\mathcal{F}_1) \\
= \; & \{w \in \mathfrak{B}^* \mid T(\mathrm{decode}(w)) \in \mathcal{F}_1\} \\
= \; & \{w \in \mathfrak{B}^* \mid T(\mathrm{decode}(w))(0) \text{ ist nicht definiert}\} \\
= \; & \{w \in \mathfrak{B}^* \mid \mathrm{decode}(w) \text{ hält nicht bei } 0\} \\
= \; & L_1
\end{aligned}$$

(b) $L_2 = \{w \in \mathfrak{B}^* \mid \mathrm{decode}(w)$ hat genau drei Zustände$\}$ Der Satz von Rice ist nicht anwendbar und dies kann wie folgt bewiesen werden: (Widerspruchsbeweis) Angenommen, es gäbe eine nichttriviale Menge $\mathcal{F}_2$ von berechenbaren partiellen Funktionen sodass $\mathcal{C}(\mathcal{F}_2) = L_2$. Sei M_1 eine Turingmaschine mit genau 3 Zuständen welche $f : \mathfrak{B}^* \to \mathfrak{B}^* : w \mapsto w$ berechnet, und sei M_2 eine Turingmaschine mit genau 4 Zuständen, welche dieselbe Funktion f berechnet. Dann gilt offensichtlich $\mathrm{code}(M_1) \in L_2$ und $\mathrm{code}(M_2) \notin L_2$, wegen $T(M_1) = T(M_2)$ aber auch $\mathrm{code}(M_1) \in \mathcal{C}(\mathcal{F}_2)$ gdw. $\mathrm{code}(M_2) \in \mathcal{C}(\mathcal{F}_2)$, Widerspruch zu $\mathcal{C}(\mathcal{F}_2) = L_2$.

(Faktisch ist L_2 auch entscheidbar, aber das muss nicht sein!)

Hausaufgabe 6.5 (Entscheidbarkeit und Berechenbarkeit) (6)

Beweisen oder widerlegen Sie die folgenden Aussagen:

(a) Die Menge $\mathcal{P}(\Sigma^*)$ aller Sprachen über Σ ist nicht rekursiv aufzählbar. Wahr. Beweis per Widerspruch. Nimm also an, dass $\mathcal{P}(\Sigma^*)$ rekursiv aufzählbar ist. Dann ist $\mathcal{P}(\Sigma^*)$ auch abzählbar ●$_1$ (siehe Vorlesung 8, Folie 33), Widerspruch zu Satz §1.13 in Vorlesung 1, wo wir gezeigt haben, dass $\mathcal{P}(\Sigma^*)$ nicht abzählbar ist●$_2$.

(b) Alle totalen Funktionen sind berechenbar. Falsch. Zum Beispiel ist die charakteristische Funktion χ_H des Halteproblems H nicht berechenbar●3, aber total●4.

(c) Jede Teilmenge einer nicht entscheidbaren Sprache L ist auch nicht entscheidbar. Falsch. Denn die leere Menge $\emptyset$ ist Teilmenge jeder Menge L●5, und $\emptyset$ ist entscheidbar●6.

Hausaufgabe 6.6 (Entscheidbarkeit und Berechenbarkeit) (7)

Seien $L_1, L_2 \subseteq \Sigma^*$ zwei rekursiv aufzählbare Sprachen über einem endlichen Alphabet Σ sodass $L_1 \cap L_2 = \emptyset$. Weiterhin sei $L = L_1 \cup L_2$. Zeigen Sie, dass L_1 entscheidbar ist, falls L entscheidbar ist.

LÖSUNG: Angenommen, L sei entscheidbar. Dann gibt es eine deterministische Turingmaschine M, welche χ_L berechnet. Das heisst, bei Eingabe von $w \in \Sigma^*$ hält M in q_+ mit Bandhinhalt 1 falls $w \in L$ und in q_+ mit Bandhinhalt 0 falls $w \notin L$. ●7 Wegen $L_1 \cap L_2 = \emptyset$ gilt $w \in L$ gdw. $w \in L_1$ und $w \notin L_2$, oder $w \in L_2$ und $w \notin L_1$. Außerdem gilt $w \notin L$ gdw. $w \notin L_1$ und $w \notin L_2$. ●8 L_1 und L_2 sind nach Annahme rekursiv aufzählbar, d.h. semientscheidbar. Es gibt also eine deterministische Turingmaschine M_1 welche die halbe charakteristische Funktion ρ_{L_1} berechnet, und es gibt eine deterministische Turingmaschine M_2 welche die halbe charakteristische Funktion ρ_{L_2} berechnet. Das heisst, falls $w \in L_1$, so hält M_1 in q_+^1 mit Bandinhalt 1 und falls $w \in L_2$, so hält M_2 in q_+^2 mit Bandinhalt 1. ●9 Wir definieren eine neue deterministische Turingmaschine M' welche χ_{L_1} berechnet: M' simuliert zunächst M auf der Eingabe w. ●10 Falls M in q_+ mit Bandinhalt 0 stoppt, also $w \notin L$, also $w \notin L_1$, so geht M' in q'_+ mit Bandinhalt 0. ●11 Falls M in q_+ mit Bandinhalt 1 stoppt (also $w \in L$), so simuliert M die beiden Turingmaschinen M_1 und M_2 schrittweise synchron ("parallel") ●12. Da $w \in L$, muss genau eine der beiden Turingmaschinen in seinem jeweiligen akzeptierenden Zustand q_+^i mit Bandinhalt 1 stoppen. Falls M_1 in q_+^1 mit Bandinhalt 1 stoppt, so geht M' in q'_+ mit Bandinhalt 1, denn dann $w \in L_1$. Falls M_2 in q_+^2 mit Bandinhalt 1 stoppt, so geht M' in q'_+ mit Bandinhalt 0, denn dann $w \in L_2$, also $w \notin L_1$. ●13

Hausaufgabe 6.7 (Reduktion) (7)

Definiere die Sprache $H_\exists \subseteq \mathfrak{B}^*$ durch

$$H_\exists = \{w \in \mathfrak{B}^* \mid \text{decode}(w) \text{ hält auf mindestens einem Wort}\}.$$

Zeigen Sie durch Reduktion dass $H_\exists$ unentscheidbar ist.

LÖSUNG: Wir zeigen $H_\varepsilon \preceq H_\exists$. Dazu definieren wir eine totale berechenbare Funktion $f : \mathfrak{B}^* \to \mathfrak{B}^*$ sodass $w \in H_\varepsilon$ gdw. $f(w) \in H_\exists$. ●14 Definiere, für alle $w \in \mathfrak{B}^*$, $f(w) = \text{code}(M_w)$, wobei M_w eine bereinigte deterministische TM ist, welche zunächst jede Eingabe vom Eingabeband löscht und dann die TM decode(w) auf leerem Band simuliert. ●15 ●16 In der Tat ist f total ●17 und berechenbar ●18. Angenommen, $w \in H_\varepsilon$. Dann hält decode(w) auf leerem Band. Dann hält M_w auf jeder Eingabe, also auch auf mindestens einem Eingabewort. Also $\text{code}(M_w) \in H_\exists$ ●[19]. Angenommen, $w \notin H_\varepsilon$. Dann hält $\text{decode}(w)$ nicht auf leerem Band. Dann hält M_w auf keiner Eingabe. Also $\text{code}(M_w) \notin H_\exists$ ●[20].

Hausaufgabe 6.8 (Satz von Rice) (10)

Für die folgenden Sprachen L_i prüfen Sie bitte, ob der Satz von Rice anwendbar ist, um die Unentscheidbarkeit von L_i zu zeigen.

(a) $L_1 = \{w \in \mathfrak{B}^* \mid \text{decode}(w) \text{ berechnet den Nachfolger der Zahl, die durch das Eingabewort } w' \text{ binär dargestellt ist}\}$ Der Satz von Rice ist anwendbar ●[21]. Setze $\mathcal{F}_1 = \{\text{nf}\}$, wobei $\text{nf} : \mathfrak{B}^* \to \mathfrak{B}^*$ die Funktion ist, welche jedes Wort auf seinen Nachfolger abbildet ●[22]. Es gilt $\emptyset \subsetneq \mathcal{F}_1$, denn z.B. $\text{id} \notin \mathcal{F}_1$ ●[23]. Weiterhin gilt $\mathcal{F}_1 \subsetneq \mathcal{R}$ denn z.B. $\perp \in \mathcal{R} \setminus \mathcal{F}_1$ ●[24]. Weiterhin gilt $\mathcal{C}(\mathcal{F}_1) = L_1$, denn

$$
\begin{aligned}
& \mathcal{C}(\mathcal{F}_1) \\
= \; & \{w \in \mathfrak{B}^* \mid T(\text{decode}(w)) \in \mathcal{F}_1\} \\
= \; & \{w \in \mathfrak{B}^* \mid T(\text{decode}(w)) = \text{nf}\} \\
= \; & L_1
\end{aligned}
$$

●[25]

(b) $L_2 = \{w \in \mathfrak{B}^* \mid \text{decode}(w) \text{ hält nach eine geraden Anzahl von Schritten bei Eingabe } 001\}$ Der Satz von Rice ist hier nicht anwendbar. ●[26] Intuitiv handelt es sich bei der Eigenschaft “hält nach einer geraden Anzahl von Schritten” um eine Eigenschaft der Berechnung, nicht der berechneten Funktion. Formal beweisen wir die Nichtanwendbarkeit wieder mit einem Widerspruchsbeweis. Angenommen, es gäbe eine nichttriviale Menge $\mathcal{F}_2$ von berechenbaren partiellen Funktionen sodass $\mathcal{C}(\mathcal{F}_2) = L_2$. ●[27] Sei M_1 eine deterministische Turingmaschine welche die partielle Funktion $f : \mathfrak{B}^* \dashrightarrow \mathfrak{B}^*$ berechnet, welche die Eingabe 001 auf 001 abbildet, und für alle anderen Eingaben undefiniert ist. Weiterhin soll M_1 bei Eingabe 001 genau 8 Schritte verwenden bis sie hält (4 Schritte für das Prüfen der Eingabe, und 4 Schritte für das Zurücklaufen zum Anfang der Eingabe). ●[28] Sei M_2 eine Turingmaschine, welche dieselbe Funktion f berechnet, aber bei Eingabe von 001 genau 9 Schritte verwendet. ●[29] (Beide Turingmaschinen existieren, wie leicht einzusehen ist.) Dann gilt offensichtlich $\text{code}(M_1) \in L_2$ und $\text{code}(M_2) \notin L_2$, wegen $T(M_1) = T(M_2)$ aber auch $\text{code}(M_1) \in \mathcal{C}(\mathcal{F}_2)$ gdw. $\text{code}(M_2) \in \mathcal{C}(\mathcal{F}_2)$, Widerspruch zu $\mathcal{C}(\mathcal{F}_2) = L_2$. ●[30]