Reduktion und der Satz von Rice

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik

Stand: 27. Juni 2023
Folien ursprünglich von PD Dr. David Sabel

Spezielles Halteproblem

Wiederholung:

Definition (Spezielles Halteproblem)

Das spezielle Halteproblem ist die Sprache

$$K := \{w \in \{0,1\}^* \mid M_w \text{ hält für Eingabe } w\}$$

Beachte: K ist semi-entscheidbar:

Wiederhole für $i = 0, 1, \ldots$:

Simuliere i Schritte von M_w auf Eingabe w.

Wenn M_w akzeptiert, dann akzeptiere und gib 1 aus.

Beachte: K unentscheidbar, aber semi-entscheidbar $\Longrightarrow$ $\overline{K}$ nicht semi-entscheidbar.

Der Beweis, dass K unentscheidbar ist, verwendet ein Diagonalisierungsargument, das wir gleich genauer erläutern.

Diagonalisierungsargument

	w_1	w_2	w_3	$\dots$
M_{w_1}	*ja*	*nein*	*ja*	$\dots$
M_{w_2}	*nein*	*nein*	*ja*	$\dots$
M_{w_3}	*ja*	*nein*	*ja*	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$

- Spalten: alle Wörter über $\{0, 1\}$: $w_1, w_2, \dots$.
- Zeilen: alle TMs $M_{w_1}, M_{w_2}, \dots$.
- Eintrag in Zeile i und Spalte j: *ja*, wenn M_{w_i} bei Eingabe w_j akzeptiert, *nein* sonst.

Diagonalisierungsargument

	w_1	w_2	w_3	$\dots$
M_{w_1}	*ja*	*nein*	*ja*	$\dots$
M_{w_2}	*nein*	*nein*	*ja*	$\dots$
M_{w_3}	*ja*	*nein*	*ja*	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
M_{w_D}				

- Spalten: alle Wörter über $\{0,1\}$: $w_1, w_2, \dots$.
- Zeilen: alle TMs $M_{w_1}, M_{w_2}, \dots$.
- Eintrag in Zeile i und Spalte j: *ja*, wenn M_{w_i} bei Eingabe w_j akzeptiert, *nein* sonst.
- Sei $L_D = \{w_i \mid M_{w_i}$ hält nicht für Eingabe $w_i\}$ $(= \overline{K})$.
- Sei w_D die TM-Beschreibung von TM M_{w_D}, die χ_{L_D} berechnet.

Diagonalisierungsargument

	w_1	w_2	w_3	$\dots$
M_{w_1}	*ja*	*nein*	*ja*	$\dots$
M_{w_2}	*nein*	*nein*	*ja*	$\dots$
M_{w_3}	*ja*	*nein*	*ja*	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
M_{w_D}	*nein*			

- Spalten: alle Wörter über $\{0,1\}$: $w_1, w_2, \dots$.
- Zeilen: alle TMs $M_{w_1}, M_{w_2}, \dots$.
- Eintrag in Zeile i und Spalte j: *ja*, wenn M_{w_i} bei Eingabe w_j akzeptiert, *nein* sonst.
- Sei $L_D = \{w_i \mid M_{w_i}$ hält nicht für Eingabe $w_i\}$ $(= \overline{K})$.
- Sei w_D die TM-Beschreibung von TM M_{w_D}, die χ_{L_D} berechnet.

Diagonalisierungsargument

	w_1	w_2	w_3	$\dots$
M_{w_1}	*ja*	*nein*	*ja*	$\dots$
M_{w_2}	*nein*	*nein*	*ja*	$\dots$
M_{w_3}	*ja*	*nein*	*ja*	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
M_{w_D}	*nein*	*ja*		

- Spalten: alle Wörter über $\{0, 1\}$: $w_1, w_2, \dots$.
- Zeilen: alle TMs $M_{w_1}, M_{w_2}, \dots$.
- Eintrag in Zeile i und Spalte j: *ja*, wenn M_{w_i} bei Eingabe w_j akzeptiert, *nein* sonst.
- Sei $L_D = \{w_i \mid M_{w_i}$ hält nicht für Eingabe $w_i\}$ $(= \overline{K})$.
- Sei w_D die TM-Beschreibung von TM M_{w_D}, die χ_{L_D} berechnet.

Diagonalisierungsargument

	w_1	w_2	w_3	$\dots$
M_{w_1}	*ja*	*nein*	*ja*	$\dots$
M_{w_2}	*nein*	*nein*	*ja*	$\dots$
M_{w_3}	*ja*	*nein*	*ja*	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
M_{w_D}	*nein*	*ja*	*nein*	

- Spalten: alle Wörter über $\{0,1\}$: $w_1, w_2, \dots$.
- Zeilen: alle TMs $M_{w_1}, M_{w_2}, \dots$.
- Eintrag in Zeile i und Spalte j: *ja*, wenn M_{w_i} bei Eingabe w_j akzeptiert, *nein* sonst.
- Sei $L_D = \{w_i \mid M_{w_i}$ hält nicht für Eingabe $w_i\}$ $(= \overline{K})$.
- Sei w_D die TM-Beschreibung von TM M_{w_D}, die χ_{L_D} berechnet.

Diagonalisierungsargument

	w_1	w_2	w_3	$\dots$	w_D
M_{w_1}	*ja*	*nein*	*ja*	$\dots$	$\dots$
M_{w_2}	*nein*	*nein*	*ja*	$\dots$	$\dots$
M_{w_3}	*ja*	*nein*	*ja*	$\dots$	$\dots$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
M_{w_D}	*nein*	*ja*	*nein*	$\dots$	

- Spalten: alle Wörter über $\{0, 1\}$: $w_1, w_2, \dots$.
- Zeilen: alle TMs $M_{w_1}, M_{w_2}, \dots$.
- Eintrag in Zeile i und Spalte j: *ja*, wenn M_{w_i} bei Eingabe w_j akzeptiert, *nein* sonst.
- Sei $L_D = \{w_i \mid M_{w_i}$ hält nicht für Eingabe $w_i\}$ $(= \overline{K})$.
- Sei w_D die TM-Beschreibung von TM M_{w_D}, die χ_{L_D} berechnet.

Diagonalisierungsargument

	w_1	w_2	w_3	$\ldots$	w_D
M_{w_1}	*ja*	*nein*	*ja*	$\ldots$	$\ldots$
M_{w_2}	*nein*	*nein*	*ja*	$\ldots$	$\ldots$
M_{w_3}	*ja*	*nein*	*ja*	$\ldots$	$\ldots$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
M_{w_D}	*nein*	*ja*	*nein*	$\ldots$	**??**

- Spalten: alle Wörter über $\{0, 1\}$: $w_1, w_2, \ldots$.
- Zeilen: alle TMs $M_{w_1}, M_{w_2}, \ldots$.
- Eintrag in Zeile i und Spalte j: *ja*, wenn M_{w_i} bei Eingabe w_j akzeptiert, *nein* sonst.
- Sei $L_D = \{w_i \mid M_{w_i}$ hält nicht für Eingabe $w_i\}$ $(= \overline{K})$.
- Sei w_D die TM-Beschreibung von TM M_{w_D}, die χ_{L_D} berechnet.
- Eintrag in Zeile M_{w_D} und Spalte w_D ist *ja*, g.d.w. der Eintrag in Spalte w_D und Zeile M_{w_D} *nein* ist. Widerspruch. L_D ist daher nicht entscheidbar.

Reduktion

- Hilfsmittel, um Unentscheidbarkeit nachzuweisen
- Statt Unentscheidbarkeit von Sprache L von grundauf neu zu beweisen, zeige:

  Wenn man L entscheiden könnte, dann könnte man auch K entscheiden.
- Da K bereits als unentscheidbar gezeigt wurde, folgt L ist unentscheidbar.
- Statt K können wir eine beliebige Sprache nehmen, die bereits als unentscheidbar bewiesen ist.

Reduktion (Definition)

Definition (Reduktion (einer Sprache auf eine andere))

Sei $L_1 \subseteq \Sigma_1^*$ und $L_2 \subseteq \Sigma_2^*$ Sprachen. Dann sagen wir L_1 ist auf L_2 reduzierbar (geschrieben $L_1 \leq L_2$), falls es eine totale berechenbare Funktion $f : \Sigma_1^* \to \Sigma_2^*$ gibt, sodass für alle $w \in \Sigma_1^*$ gilt: $w \in L_1 \iff f(w) \in L_2$.

Reduktion (Definition)

Definition (Reduktion (einer Sprache auf eine andere))

Sei $L_1 \subseteq \Sigma_1^*$ und $L_2 \subseteq \Sigma_2^*$ Sprachen. Dann sagen wir L_1 ist auf L_2 reduzierbar (geschrieben $L_1 \leq L_2$), falls es eine totale berechenbare Funktion $f : \Sigma_1^* \to \Sigma_2^*$ gibt, sodass für alle $w \in \Sigma_1^*$ gilt: $w \in L_1 \iff f(w) \in L_2$.

Eselsbrücke:

$$\underbrace{L_1}_{\text{„kleines“ Problem}} \leq \underbrace{L_2}_{\text{„großes“ Problem}}$$

Reduktion (Definition)

Definition (Reduktion (einer Sprache auf eine andere))

Sei $L_1 \subseteq \Sigma_1^*$ und $L_2 \subseteq \Sigma_2^*$ Sprachen. Dann sagen wir L_1 ist auf L_2 reduzierbar (geschrieben $L_1 \leq L_2$), falls es eine totale berechenbare Funktion $f : \Sigma_1^* \to \Sigma_2^*$ gibt, sodass für alle $w \in \Sigma_1^*$ gilt: $w \in L_1 \iff f(w) \in L_2$.

Eselsbrücke:

$$\underbrace{L_1}_{\text{„kleines" Problem}} \leq \underbrace{L_2}_{\text{„großes" Problem}}$$

- $\leq$ sagt die Wahrheit.
- „Reduktion" täuscht.
  Man reduziert das kleine Problem auf das große.
- Das kleine Problem sieht nicht immer kleiner aus.

Entscheidbarkeit

Satz

Wenn $L_1 \leq L_2$ und L_2 (semi-)entscheidbar ist, dann ist auch L_1 (semi-)entscheidbar.

Entscheidbarkeit

Satz

Wenn $L_1 \leq L_2$ und L_2 (semi-)entscheidbar ist, dann ist auch L_1 (semi-)entscheidbar.

Beweis (nur entscheidbar, semi-entscheidbar analog): Sei f die $L_1 \leq L_2$ bezeugende Funktion. Da L_2 entscheidbar ist, ist χ_{L_2} berechenbar. Es gilt

$$\chi_{L_1}(w) = 1 \iff w \in L_1 \iff f(w) \in L_2 \iff \chi_{L_2}(f(w)) = 1$$

Damit ist $\chi_{L_1}(w) = \chi_{L_2}(f(w))$ berechenbar. □

Unentscheidbarkeit

Mit Kontraposition folgt:

Lemma

Sei $L_1 \leq L_2$ und L_1 ist unentscheidbar.
Dann ist auch L_2 unentscheidbar.

Das ist die Richtung, die wir meistens brauchen:

- L_1 sei eine bekannt unentscheidbare Sprache
- Reduziere L_1 auf L_2 durch Angabe einer berechenbaren Funktion f mit $w \in L_1 \iff f(w) \in L_2$.
- Damit folgt, dass L_2 unentscheidbar ist.

Halteproblem

Definition (Halteproblem)

Das (allgemeine) Halteproblem ist $H := \{w\#x \mid \text{TM } M_w \text{ hält für Eingabe } x\}$

Halteproblem

Definition (Halteproblem)

Das (allgemeine) Halteproblem ist $H := \{w\#x \mid \text{TM } M_w \text{ hält für Eingabe } x\}$

Satz

Das allgemeine Halteproblem ist unentscheidbar.

Halteproblem

Definition (Halteproblem)

Das (allgemeine) Halteproblem ist $H := \{w\#x \mid \text{TM } M_w \text{ hält für Eingabe } x\}$

Satz

Das allgemeine Halteproblem ist unentscheidbar.

Beweis: Wir reduzieren das spezielle Halteproblem („das kleine Problem“) auf das allgemeine Halteproblem („das große Problem“), und zeigen daher $K \leq H$. Sei $f(w) = w\#w$. Dann gilt

$$\begin{aligned} & w \in K \\ \text{g.d.w.}\quad & M_w \text{ hält für Eingabe } w \\ \text{g.d.w.}\quad & w\#w \in H \\ \text{g.d.w.}\quad & f(w) \in H \end{aligned}$$

f kann durch eine TM berechnet werden. Damit gilt $K \leq H$ und damit folgt aus K unentscheidbar auch H unentscheidbar. $\square$

Halteproblem bei leerer Eingabe

Definition (Halteproblem auf leerem Band)

Das Halteproblem auf leerem Band ist $H_0 = \{w \mid M_w \text{ hält für die leere Eingabe}\}$.

Halteproblem bei leerer Eingabe

Definition (Halteproblem auf leerem Band)

Das Halteproblem auf leerem Band ist $H_0 = \{w \mid M_w \text{ hält für die leere Eingabe}\}$.

Satz

Das Halteproblem auf leerem Band ist unentscheidbar.

Halteproblem bei leerer Eingabe

Definition (Halteproblem auf leerem Band)

Das Halteproblem auf leerem Band ist $H_0 = \{w \mid M_w \text{ hält für die leere Eingabe}\}$.

Satz

Das Halteproblem auf leerem Band ist unentscheidbar.

Beweis: Wir reduzieren H auf H_0: Sei $f(w_M\#x) = w_{M_{0,x}}$, wobei TM $M_{0,x}$ erst x auf das Band schreibt, sich dann wie M verhält.

$$
\begin{aligned}
& w_M\#x \in H \\
\text{g.d.w.}\quad & M \text{ hält für Eingabe } x \\
\text{g.d.w.}\quad & M_{0,x} \text{ hält für die leere Eingabe} \\
\text{g.d.w.}\quad & w_{M_{0,x}} \in H_0 \\
\text{g.d.w.}\quad & f(w_M\#x) \in H_0
\end{aligned}
$$

f kann durch eine TM berechnet werden. Daher gilt $H \leq H_0$. Da H unentscheidbar ist, ist H_0 unentscheidbar. □

Der Satz von Rice

Von Henry Gordon Rice im Jahr 1953 veröffentlicht.

Satz von Rice

Sei $\mathcal{R}$ die Klasse aller Turingberechenbaren Funktionen. Sei $\mathcal{S}$ eine beliebige Teilmenge, sodass $\emptyset \subset \mathcal{S} \subset \mathcal{R}$. Dann ist folgende Sprache unentscheidbar:

$$C(\mathcal{S}) = \{w \mid \text{die von } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\}$$

Der Satz zeigt:

- Fast alle interessanten Eigenschaften von TMs sind algorithmisch nicht entscheidbar.
- Z.B. folgt, dass die Sprache

  $$L = \{w \mid M_w \text{ berechnet eine konstante Funktion}\}$$

  nicht entscheidbar ist.

Beweis des Satzes von Rice (1)

Sei $\Omega(x) =$ undefiniert für alle x.

Zeige:

1. $H_0 \leq C(\mathcal{S})$ falls $\Omega \notin \mathcal{S}$.
2. $H_0 \leq C(\mathcal{S})$ falls $\Omega \in \mathcal{S}$.

Wir beweisen nur 1), da 2) analog geht (siehe Skript).

Beweis des Satzes von Rice (2)

Fall: $\Omega \notin \mathcal{S}$. Da $\emptyset \subset \mathcal{S}$, gibt es eine Funktion $q \in \mathcal{S}$, die von einer TM Q berechnet wird.

Konstruktion einer TM M^*: Für TM M und Eingabe y:

1. M^* simuliert M auf leerer Eingabe.
2. Wenn M anhält, dann simuliert M^* die TM Q mit Eingabe y.

Sei f die Funktion, die aus der Beschreibung w für TM M_w, die Beschreibung $f(w)$ von M_w^* erstellt. Diese Funktion ist total und berechenbar.

Dann gilt: $w \in H_0 \implies M_w$ hält auf leerer Eingabe
$\implies M_w^*$ berechnet q
$\implies$ die von M_w^* berechnete Funktion liegt in $\mathcal{S}$
$\implies f(w) \in C(\mathcal{S})$

und ebenso: $w \notin H_0 \implies M_w$ hält nicht auf leerer Eingabe
$\implies M_w^*$ berechnet Ω
$\implies$ die von M_w^* berechnete Funktion liegt nicht in $\mathcal{S}$
$\implies f(w) \notin C(\mathcal{S})$

Beweis des Satzes von Rice (3)

Aus $w \in H_0 \implies f(w) \in C(\mathcal{S})$ und $w \notin H_0 \implies f(w) \notin C(\mathcal{S})$ folgt

$$w \in H_0 \Longleftrightarrow f(w) \in C(\mathcal{S})$$

Daher $H_0 \leq C(\mathcal{S})$.

Da H_0 unentscheidbar ist, ist damit auch $C(\mathcal{S})$ unentscheidbar. $\square$

Beispiele für die Anwendung des Satzes von Rice

Beispiel

Es ist unentscheidbar, ob eine Turingmaschine für jede Eingabe $i \in \mathbb{N}$ die Zahl $i + 1 \in \mathbb{N}$ berechnet.

- Sei $succ(i) = i + 1$.
- Sei $\mathcal{S} := \{succ\}$.
- $\mathcal{S}$ ist nicht trivial:
  - $\emptyset \subset \mathcal{S}$: klar
  - $\mathcal{S} \subset \mathcal{R}$: f mit $f(i) = i + 2$ ist berechenbar, aber $f \notin \mathcal{S}$.
- Mit Satz von Rice:

  $$C(\mathcal{S}) = \{w \mid \text{die von } M_w \text{ berechnete Funktion ist } succ\}$$

  ist nicht entscheidbar.

Beispiele für die Anwendung des Satzes von Rice (2)

Beispiel

Es ist unentscheidbar, ob für Turingmaschine M gilt, dass $L(M) = \emptyset$.

- Sei $\mathcal{S} := \{\Omega\}$.
- $\mathcal{S}$ ist nicht trivial:
  - $\emptyset \subset \mathcal{S}$: klar
  - $\mathcal{S} \subset \mathcal{R}$: f mit $f(x) = x$ ist berechenbar, aber $f \notin \mathcal{S}$.
- Mit Satz von Rice:

$$\begin{aligned} C(\mathcal{S}) &= \{w \mid \text{die von } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\} \\ &= \{w \mid M_w \text{ akzeptiert nie}\} \\ &= \{w \mid L(M_w) = \emptyset\} \end{aligned}$$

ist nicht entscheidbar.

Beispiele für die Anwendung des Satzes von Rice (3)

Beispiel

Es ist unentscheidbar, ob für Turingmaschine M gilt, dass M für alle Eingaben hält.

- Sei $\mathcal{S} := \{f \mid f(x) \text{ ist total und berechenbar}\}$.
- $\mathcal{S}$ ist nicht trivial:
  - $\emptyset \subset \mathcal{S}$: Z.B. gilt $id \in \mathcal{S}$ mit $id(x) = x$ für alle $x \in \mathbb{N}$.
  - $\mathcal{S} \subset \mathcal{R}$: $f(1) = \text{undefiniert}$ und $f(x) = 0$ für $x \neq 1$, ist berechenbar und $f \notin \mathcal{S}$.
- Mit Satz von Rice:

$$\begin{aligned} C(\mathcal{S}) &= \{w \mid \text{die von } M_w \text{ berechnete Funktion liegt in } \mathcal{S}\} \\ &= \{w \mid M_w \text{ akzeptiert für jede Eingabe}\} \end{aligned}$$

  ist nicht entscheidbar.

Bemerkung

Beachte:

- Der Satz von Rice lässt sich anwenden, auf Eigenschaften von $L(M)$ bzw. der von M berechneten Funktion, **aber**:
- Der Satz von Rice macht keine Aussage über Eigenschaften von M.

Beispiele:

- Ist es entscheidbar, ob M höchstens 100 Zustände hat?
  ☞ Satz von Rice ist hier **nicht** anwendbar (sogar entscheidbar).
- Ist es entscheidbar, ob M für jede Eingabe nach 1000 Schritten anhält?
  ☞ Satz von Rice ist hier **nicht** anwendbar (sogar entscheidbar)
- Ist es entscheidbar, ob M für höchstens 50 verschiedene Eingaben anhält?
  ☞ Satz von Rice ist anwendbar, da die Eigenschaft auch etwas über die berechnete Funktion aussagt (für höchstens 50 Eingaben definiert). (Problem ist auch unentscheidbar.)