

Vorlesungsmitschrift

Algorithmen und Berechenbarkeit

Vorlesung 15

Letztes Update: 2018/01/16 - 12:15 Uhr

Satz von Rice

TMs halten nicht auf jeder Eingabe, das bedeutet, sie berechnen partielle Funktionen, die die folgende Form aufweisen:

$$f : \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\perp\}$$

$\rightarrow \perp$ bedeutet, TM hält nicht

Im Rahmen der Vorlesung liegt der Hauptfokus auf TMs, die JA/NEIN ausgeben, also Funktionen der Form:

$$f : \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\perp\}$$

$\rightarrow 0$ TM verwirft Eingabe
 $\rightarrow 1$ TM akzeptiert Eingabe

Sei nun die Menge aller Funktionen, die von irgendeiner TM erkannt werden, definiert als

$$\mathcal{R} = \{f_{\mathcal{M}} : \{0, 1\}^* \rightarrow \{0, 1\}^* \cup \{\perp\} \mid \mathcal{M} \text{ ist TM}\}$$

dann besagt der **Satz von Rice**:

Sei \mathcal{S} eine Teilmenge von \mathcal{R} mit $\emptyset \neq \mathcal{S} \neq \mathcal{R}$, dann ist die Sprache

$$\mathcal{L}(\mathcal{S}) = \{\langle M \rangle \mid \mathcal{M} \text{ berechnet eine Funktion aus } \mathcal{S}\}$$

unentscheidbar. □

Beweis: Man nimmt an, eine TM $\mathcal{M}_{\mathcal{R}(\mathcal{S})}$ existiert, die $\mathcal{L}(\mathcal{S})$ entscheidet. Dann konstruiert man daraus eine TM \mathcal{M}_{ϵ} , die \mathcal{H}_{ϵ} entscheidet, was im Widerspruch zur Unentscheidbarkeit von \mathcal{H}_{ϵ} steht.

Sei u die überall undefinierte Funktion. Man unterscheidet nun die Fälle

- a) $u \notin \mathcal{S}$
- b) $u \in \mathcal{S}$

Sei nun $f \neq u$ eine Funktion aus \mathcal{S} . Die TM \mathcal{M}_ϵ arbeitet wie folgt:

1. Falls die Eingabe $\langle M \rangle$ keine korrekte Kodierung einer TM ist, wird sie verworfen.
2. Es wird eine TM \mathcal{M}^* mit folgendem Verhalten konstruiert:
 - Man ignoriert die Eingabe x , d.h. man simuliert \mathcal{M} auf der Eingabe ϵ .
 - Man berechnet $f(x)$, d.h. man simuliert \mathcal{M}_f auf der Eingabe x . \Rightarrow Beobachtung: \mathcal{M}^* berechnet genau dann $f(x)$, falls \mathcal{M} auf ϵ hält. Sonst gilt $f_{\mathcal{M}} = u$.
3. Man startet die TM $\mathcal{M}_{\mathcal{L}(\mathcal{S})}$ auf der Eingabe $\langle M^* \rangle$ und akzeptiert genau dann, wenn $\mathcal{M}_{\mathcal{L}(\mathcal{S})}$ akzeptiert.

Korrektheit der Konstruktion:

- $w \in \mathcal{H}_\epsilon$
 - $\Rightarrow \mathcal{M}$ hält auf ϵ
 - $\Rightarrow \langle M^* \rangle \in \mathcal{L}(\mathcal{S})$
 - $\Rightarrow \mathcal{M}_{\mathcal{L}(\mathcal{S})}$ akzeptiert $\langle M^* \rangle$
 - $\Rightarrow \mathcal{M}_\epsilon$ akzeptiert w
- $w \notin \mathcal{H}_\epsilon$
 - $\Rightarrow \mathcal{M}$ hält nicht auf ϵ
 - $\Rightarrow \mathcal{M}^*$ berechnet u
 - $\Rightarrow \langle M^* \rangle \notin \mathcal{L}(\mathcal{S})$
 - $\Rightarrow \mathcal{M}_{\mathcal{L}(\mathcal{S})}$ verwirft $\langle M^* \rangle$
 - $\Rightarrow \mathcal{M}_\epsilon$ verwirft w

Analog kann der Fall $u \in \mathcal{S}$ gezeigt werden (mit Invertierung des Akzeptanzverhaltens). \square

Als Folge des Satzes von Rice gilt insbesondere, dass man nicht entscheiden kann, ob ein Programm auf jeder Eingabe hält.

$$\mathcal{S} = \{f : \{0, 1\}^* \rightarrow \{0, 1\}^*\}$$

Semi-Entscheidbarkeit

Definition Entscheidbarkeit: Eine Sprache L wird von einer TM \mathcal{M} **entschieden**, falls \mathcal{M} auf jeder Eingabe hält und genau die Wörter aus L akzeptiert.

Definition Erkennen: Eine Sprache L wird von einer TM \mathcal{M} **erkannt**, wenn \mathcal{M} jedes Wort aus L akzeptiert und \mathcal{M} kein Wort akzeptiert, das nicht in L ist. Auf Eingaben, die nicht in L sind, muss \mathcal{M} nicht halten (Entweder NEIN sagen, oder ewig loopen).

Definition Semi-Entscheidbarkeit: Eine Sprache heißt **semi-entscheidbar**, wenn es eine TM \mathcal{M} gibt, die L erkennt.

Bsp: Das Halteproblem ($\mathcal{H} = \{\langle M \rangle w \mid \text{TM } \mathcal{M} \text{ hält auf Eingabe } w\}$) ist semi-entscheidbar, denn es gilt

- Wenn die Eingabe nicht der Form $\langle M \rangle w$ entspricht, wird verworfen,
- Man simuliert \mathcal{M} auf w :

- Falls \mathcal{M} hält, wird akzeptiert.
- Falls \mathcal{M} nicht hält, wird ewig weiter simuliert.

Wenn eine Sprache nicht semi-entscheidbar ist, dann kann sie auch nicht entscheidbar sein. Alle Sprachen, die entscheidbar sind, sind automatisch auch semi-entscheidbar.

Rekursive Aufzählbarkeit

Definition Aufzähler: Ein Aufzähler für eine Sprache L ist eine TM mit Drucker (Ausgabeband, auf dem der Kopf nur nach rechts bewegt werden darf). Ein Aufzähler für L gibt alle Wörter aus L auf dem Drucker aus:

- Es werden nur Wörter aus L ausgedruckt.
- Jedes Wort aus L wird irgendwann ausgedruckt.

Definition Rekursiv aufzählbar: Eine Sprache L heißt rekursiv aufzählbar, wenn es einen Aufzähler für L gibt.

Satz: Eine Sprache L ist genau dann semi-entscheidbar, wenn L rekursiv aufzählbar ist.

Beweis:

„aufzählbar \Rightarrow semi-entscheidbar“:

Man konstruiert eine TM \mathcal{M} , die auf einer separaten Spur den Aufzähler simuliert. Immer wenn ein Wort ausgegeben wird, wird es mit der Eingabe verglichen. Man akzeptiert, falls die Eingabe einem aufgezählten Wort entspricht. \mathcal{M} erkennt sonst L , da jedes Wort aus L irgendwann aufgezählt wird.

„semi-entscheidbar \Rightarrow aufzählbar“:

Seien $w_1, w_2, w_3 \dots$ alle Wörter aus Σ^* in kanonischer Reihenfolge (z.B. lexikografisch) sortiert. Für $i = 1, 2, 3, \dots$ simuliert man nun i -Schritte von \mathcal{M} auf w_1, w_2, \dots, w_i . Wird dabei ein Wort akzeptiert, druckt man es aus. Offensichtlich werden nur Wörter aus L ausgedruckt. Sei $w = w_k \in L$, dann wird w_k von \mathcal{M} nach t_k Schritten akzeptiert. Wenn $i = \max(k, E_k)$, dann wird w_k betrachtet und \mathcal{M} lange genug simuliert, um w_k zu akzeptieren $\Rightarrow w_k$ wird gedruckt. \square

Eigenschaften entscheidbarer / semi-entscheidbarer Sprachen

Satz:

- Wenn die Sprachen L_1 und L_2 entscheidbar sind, so ist auch $L_1 \cap L_2$ entscheidbar.
- Wenn die Sprachen L_1 und L_2 semi-entscheidbar sind, so ist auch $L_1 \cap L_2$ semi-entscheidbar.

Satz:

- Wenn die Sprachen L_1 und L_2 entscheidbar sind, so ist auch $L_1 \cup L_2$ entscheidbar.
- Wenn die Sprachen L_1 und L_2 semi-entscheidbar sind, so ist auch $L_1 \cup L_2$ semi-entscheidbar.

Satz: Wenn L entscheidbar ist, so ist auch \overline{L} entscheidbar.

Beweis: Man invertiert die Ausgabe des Entscheiders für L .

Satz: Sind L und \overline{L} semi-entscheidbar, so ist L entscheidbar.

Beweis: Man simuliert Semi-Entscheider für L und \overline{L} parallel. Die Entscheidung ist klar, sobald einer der Entscheider akzeptiert. Einer davon muss akzeptieren, denn es gilt: $w \in L$ oder $w \notin L$.

Korollar: L ist unentscheidbar genau dann, wenn mindestens einer der beiden Sprachen L und \overline{L} nicht semi-entscheidbar ist.

Bsp: Sei \mathcal{H} semi-entscheidbar. Dann gilt: $\overline{\mathcal{M}}$ ist nicht semi-entscheidbar oder aufzählbar.

Technik der Reduktion

Das Ziel ist es, mithilfe von bereits untersuchten Sprachen zu zeigen, ob eine Sprache L entscheidbar oder semi-entscheidbar ist. Die Reduktion bildet also die Eingaben eines Problems auf Eingaben eines anderen Problems ab.

Eingabe-Eingabe-Reduktion

Definition: Es seien L_1 und L_2 Sprachen über Σ^* . Dann heißt L_1 auf L_2 reduzierbar ($L_1 \leq L_2$), wenn es eine berechenbare Funktion $f : \Sigma^* \rightarrow \Sigma^*$ gibt, sodass für alle $x \in \Sigma^*$ gilt:

$$x \in L_1 \quad \Leftrightarrow \quad f(x) \in L_2$$

Aus „Berechenbarkeitssicht“ bedeutet das insbesondere, dass L_1 nicht schwieriger zu untersuchen ist als L_2 .

Lemma: Falls $L_1 \leq L_2$ und L_2 entscheidbar, so ist auch L_1 entscheidbar.

Lemma: Falls $L_1 \leq L_2$ und L_1 unentscheidbar, so ist auch L_2 unentscheidbar.

Beweis: Man konstruiert eine TM \mathcal{M}_1 , die L_1 entscheidet unter Nutzung der TM \mathcal{M}_2 , die L_2 entscheidet. Nun berechnet man mit der Eingabe x die Funktion $f(x)$ und simuliert \mathcal{M}_2 auf $f(x)$ wobei man deren Akzeptanzverhalten übernimmt.

Dann gilt

$$\begin{aligned} \mathcal{M}_1 \text{ akzeptiert} &\Leftrightarrow \mathcal{M}_2 \text{ akzeptiert } f(x) \\ &\Leftrightarrow f(x) \in L_2 \\ &\Leftrightarrow x \in L_1 \end{aligned}$$

□

Aus dem Lemma folgt außerdem: Falls $L_1 \leq L_2$ und L_1 unentscheidbar $\Rightarrow L_2$ unentscheidbar.

Korollar: Eine Sprache \mathcal{L} ist genau dann unentscheidbar, wenn entweder \mathcal{L} oder $\overline{\mathcal{L}}$ nicht semi-entscheidbar sind.