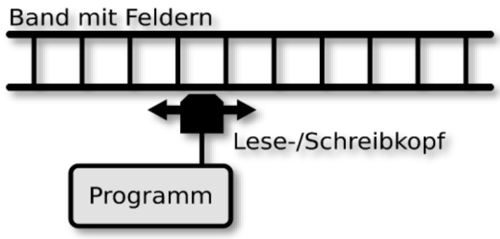
Technische Universität Braunschweig

Theoretische Informatik 2

Arne Schmidt

Kapitel 4 - Unentscheidbarkeit

Was bisher geschah...



Sei $A \subseteq \Sigma^*$ eine Sprache. Dann sind folgende Aussagen äquivalent:

1. A ist semi-entscheidbar.
2. A ist rekursiv aufzählbar.
3. Es gibt eine TM M mit $\mathcal{L}(M) = A$.
4. Es gibt einen Aufzählungsalgorithmus für A, d.h. A ist der Wertebereich einer totalen berechenbaren Funktion $f: \mathbb{N} \to \Sigma^*$.
5. χ'_A ist berechenbar.
6. A ist der Definitionsbereich einer partiellen berechenbaren Funktion $g: \Sigma_1^* \to_p \{1\}$.

Turing-Maschinen.

- Restriktion auf $\{0,1\}$ Symbole
- Restriktion auf 1-Band

LBA erkennen CSL

TM erkennen Typ-0 Sprachen

- Diagonalargument
- (Über-)abzählbar unendlich
- Es gibt nicht-berechenbare Funktionen.

Unentscheidbarkeit

Es gibt entscheidbare und semi-entscheidbare Sprachen.

Gibt es unentscheidbare Sprachen?

Gibt es auch semi-entscheidbare Sprachen, die unentscheidbar sind?

Unentscheidbarkeit

Es gibt entscheidbare und semi-entscheidbare Sprachen.

Beweis ähnlich zu nicht-berechenbaren Funktionen

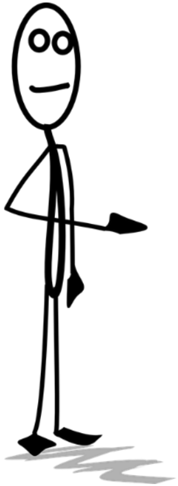
Gibt es unentscheidbare Sprachen?

Gibt es auch semi-entscheidbare Sprachen, die unentscheidbar sind?

Betrachten wir ein mal ein Problem, welches semi-entscheidbar ist.

Akzeptanzproblem

Gegeben: Eine Turing-Maschine M und eine Eingabe x.
Frage: Akzeptiert M Eingabe x, d.h. gilt $x \in \mathcal{L}(M)$?

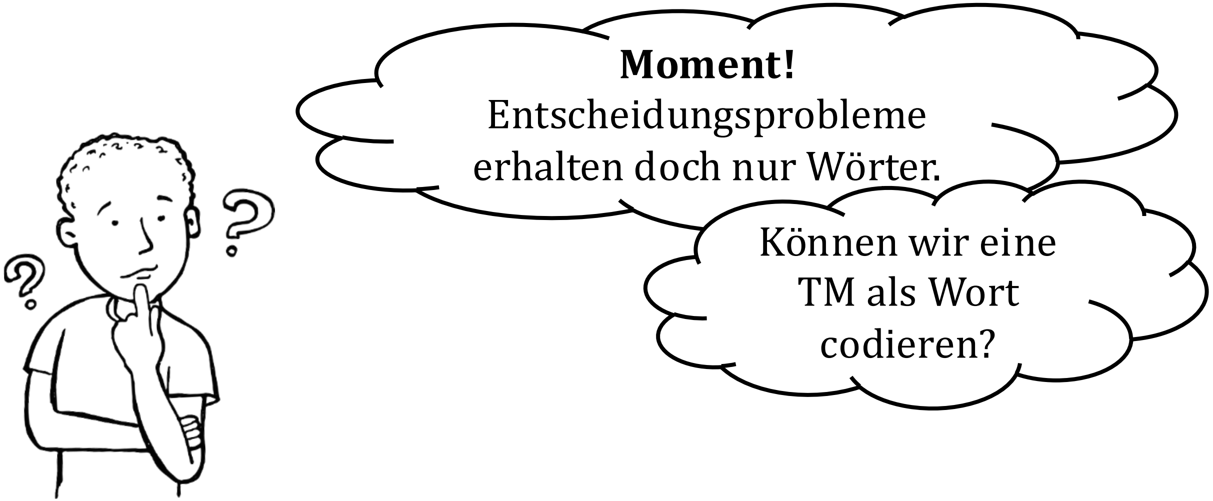


Solche Probleme werden auch Verifikationsprobleme genannt.

Ein Programm selbst ist Teil des Inputs.

Akzeptanzproblem

Gegeben: Eine Turing-Maschine M und eine Eingabe x.
Frage: Akzeptiert M Eingabe x, d.h. gilt $x \in \mathcal{L}(M)$?

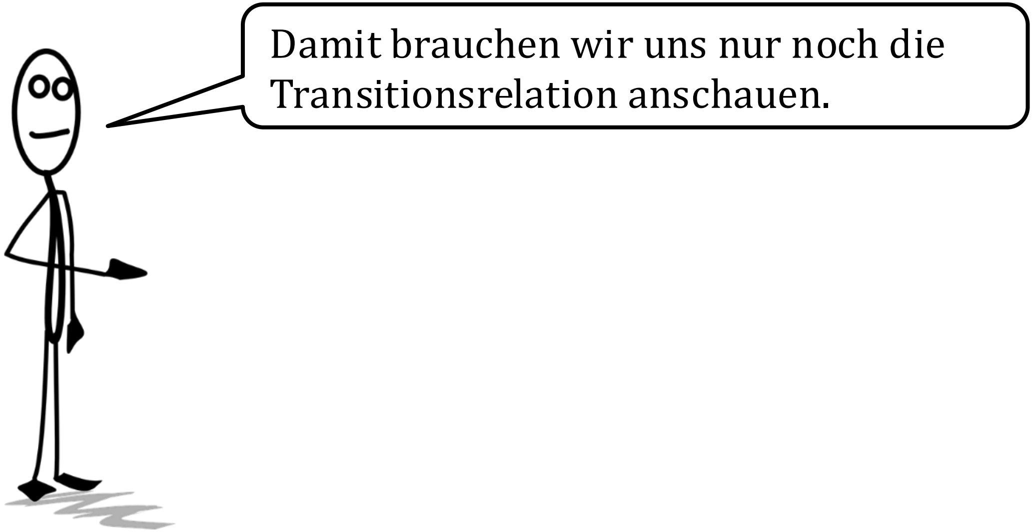


Kapitel 4.1 – Codierung von Turing-Maschinen

Codierung von Turing-Maschinen

Wir codieren eine Turing-Maschine $M = (Q, \Sigma, \Gamma, q_0, \delta, q_{acc}, q_{rej})$ mit folgenden Annahmen:

- M ist deterministisch
- M ist eine 1-Band-Maschine
- $Q = (q_0, q_1, q_2, \dots)$ mit
  - q_0 Startzustand
  - $q_1 = q_{acc}$
  - $q_2 = q_{rej}$
- $\Sigma = \{0,1\}$
- $\Gamma = \{\mathrm{a}_0, \mathrm{a}_1, \mathrm{a}_2, \dots\}$ mit
  - $a_0 = \sqcup$
  - $a_1 = 1$
  - $a_2 = 0$



Codierung der Transitionsrelation

$$\boxed{\delta(q_i, a_j) = (q_{i'}, a_{j'}, d)}$$

$$y = \begin{cases} 0, & \text{falls } d = L \\ 1, & \text{falls } d = R \\ 2, & \text{falls } d = N \end{cases}$$

$$\boxed{v_{i,i',j,j'} = \#\#\mathrm{bin}(i)\#\mathrm{bin}(j)\#\mathrm{bin}(i')\#\mathrm{bin}(j')\#\mathrm{bin}(y) \in \{0,1,\#\}^*}$$

Damit ist die gesamte Codierung:

$$v_M = \prod_{i=0}^{n} \prod_{j=0}^{m} v_{i,i',j,j'}$$

Um eine reine Binärcodierung zu erhalten, wende Homomorphismus h an:

$$0 \mapsto 00, \qquad 1 \mapsto 01, \qquad \# \mapsto 11$$

Codierung von TMs

Definition 4.1
Zu jeder TM M sei $\langle M \rangle \in \{0,1\}^*$ das Wort $h(v_m)$, welches wie vorher berschrieben entsteht.

Bemerkung 4.2
Ist q_0 ein akzeptierender / abweisender Zustand, können wir die Maschine leicht umbauen!

Bemerkung 4.3
Ist die Codierung zweier TMs gleich, so ist deren Verhalten gleich und akzeptieren dieselbe Sprache!

Wir können nun jede TM als Binärstring definieren, wir müssen aber auch aus jedem Binärstring eine TM konstruieren!

Leere TMs

Betrachte $M_\emptyset = (\{q_0, q_{acc}, q_{rej}\}, \{0,1\}, \{\sqcup, 0,1\}, \delta, q_0, q_{acc}, q_{rej})$ mit

$$\delta(q_0, a) = (q_{rej}, a, N), \qquad \forall a \in \{\sqcup, 0,1\},$$
$$\delta(q, a) = (q, a, N), \qquad q \in \{q_{acc}, q_{rej}\}, \forall a \in \{\sqcup, 0,1\}$$

Was akzeptiert die TM?
Was weist sie ab?

Sie weist jedes Wort nach einem Schritt direkt ab!
→ $\mathcal{L}(M_\emptyset) = \emptyset$

Akzeptanzproblem – Revisited

Definition 4.4
Zu jedem Wort $w \in \{0,1\}^*$ definieren wir M_w als die TM:

$$M_w = \begin{cases} M \text{ mit } \langle M \rangle = w, & \text{falls } w \text{ valide Kodierung einer TM } M \text{ ist.} \\ M_\emptyset, & \text{sonst} \end{cases}$$

Wir können das Akzeptanzproblem neu definieren, welches nun nur noch Binärstrings erhält

Gegeben: Zwei Binärstrings $w, x \in \{0,1\}^*$.
Frage: Akzeptiert M_w Eingabe x, d.h. gilt $x \in \mathcal{L}(M_w)$?

Mit Trennsymbol # erhalten wir die gewünschte Charakterisierung des Akzeptanzproblems als Sprache:

$$\text{ACCEPT} = \{\, w\#x \in \{0,1\}^*\#\{0,1\}^* \mid x \in \mathcal{L}(M_w) \,\}$$

Kapitel 4.2 – Universelle TMs

Universelle Turing-Maschinen

Sei U eine TM mit $\mathcal{L}(U) = \text{ACCEPT}$.
Wir nennen U eine **universelle Turing-Maschine (UTM)**.

Theorem 4.6
Man kann eine **universelle Turing-Maschine (UTM)** U konstruieren mit $\mathcal{L}(U) = \text{ACCEPT}$.

Simuliere die codierte Turing-Maschine mit Hilfe zusätzlicher Bänder: Programm-, Daten-, Zustandsband. Den Beweis lassen wir zunächst aus.

Theorem 4.9
Das Akzeptanzproblem ACCEPT ist semi-entscheidbar, aber nicht entscheidbar.

Semi-Entscheidbarkeit ist durch Th. 4.6 abgedeckt. Zeigen wir nun, dass es nicht entscheidbar ist!

Kapitel 4.3 – Unentscheidbarkeit des Akzeptanzproblems

Unentscheidbarkeit

Theorem 4.9
Das Akzeptanzproblem ACCEPT ist semi-entscheidbar, aber nicht entscheidbar.

Spezielles Akzeptanzproblem

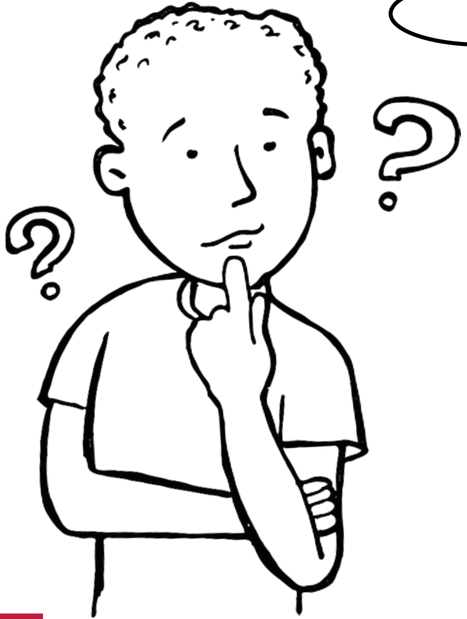
Gegeben: Eine Turing-Maschine M.
Frage: Akzeptiert M Eingabe $\langle M \rangle$?

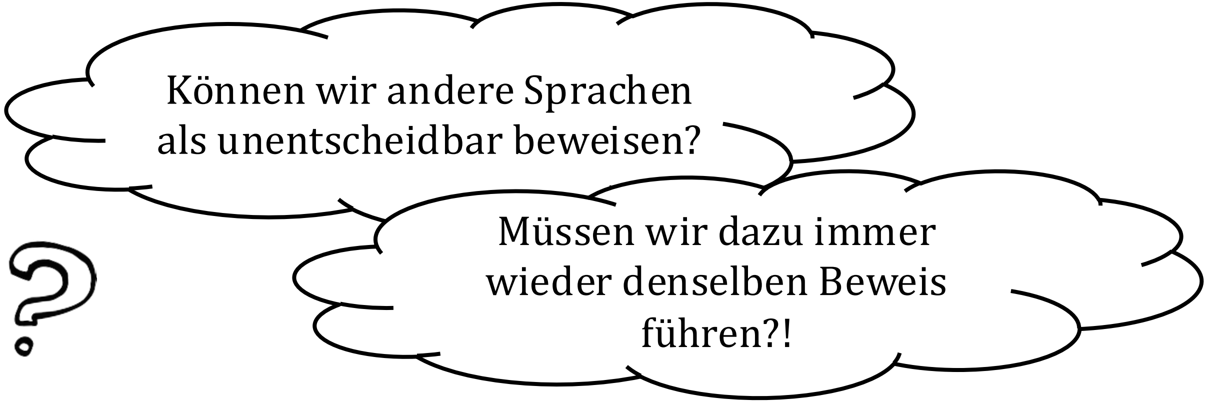
Oder als Sprache:

$$\text{SELF-ACCEPT} = \{\, w \in \{0,1\}^* \mid w \in \mathcal{L}(M_w) \,\}$$

Korollar 4.11
Das Spezielle Akzeptanzproblem SELF-ACCEPT ist semi-entscheidbar, aber nicht entscheidbar.

Nächstes Mal





Nächstes Mal: Reduktionen.
„Wenn Sprache $\mathcal{L}$ entscheidbar ist, dann muss auch ACCEPT entscheidbar sein.“