



Technische
Universität
Braunschweig

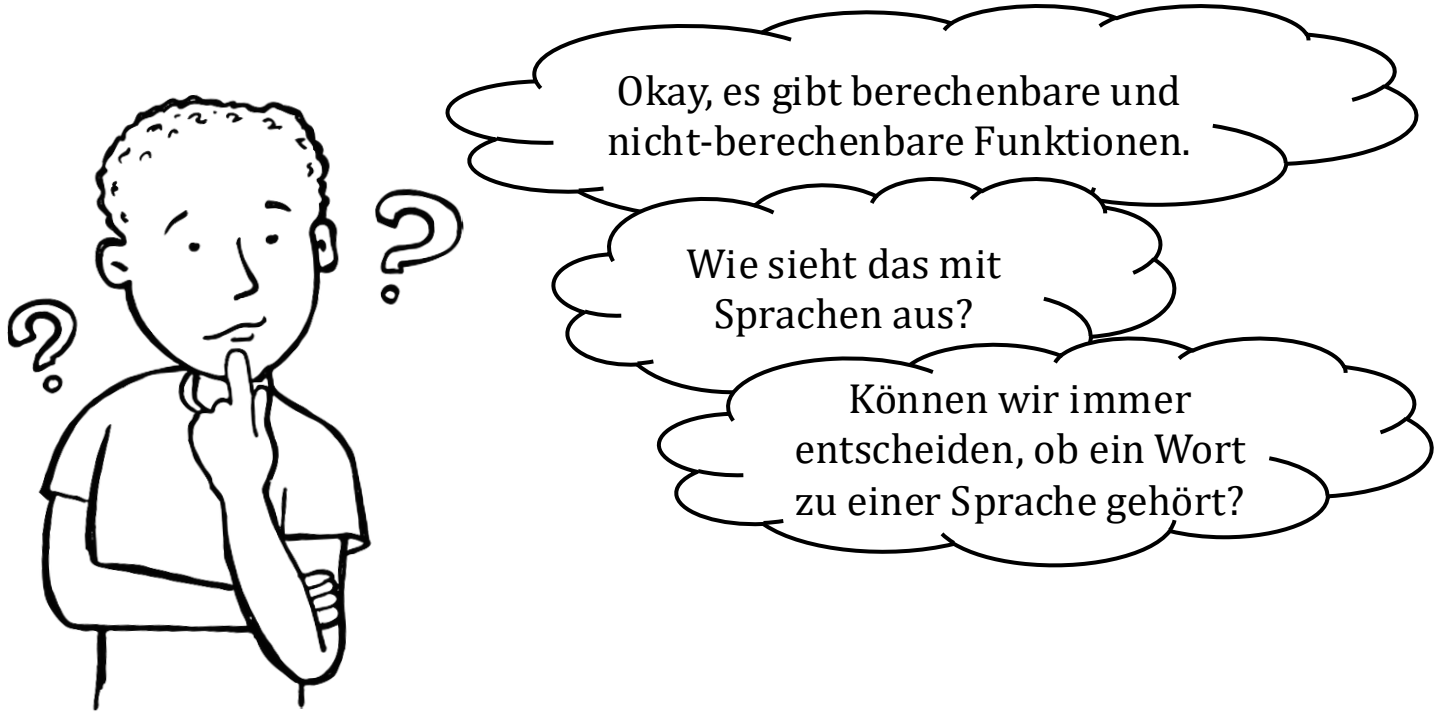


Theoretische Informatik 2

Arne Schmidt

Kapitel 3.2 – Entscheidbarkeit

Entscheidbarkeit



(Semi-)Entscheidbarkeit

Definition 3.14

Eine Menge $A \subseteq \Sigma^*$ ist **entscheidbar**, wenn die totale charakteristische Funktion χ_A von A mit

$$\chi_A: \Sigma^* \rightarrow \{0, 1\}$$
$$w \mapsto \begin{cases} 1, & \text{falls } w \in A \\ 0, & \text{sonst} \end{cases}$$

berechenbar ist.

Eine Menge A ist **semi-entscheidbar**, wenn die partielle charakteristische Funktion χ'_A von A mit

$$\chi'_A: \Sigma^* \rightarrow_p \{1\}$$
$$w \mapsto \begin{cases} 1, & \text{falls } w \in A \\ \text{undef}, & \text{sonst} \end{cases}$$

berechenbar ist.

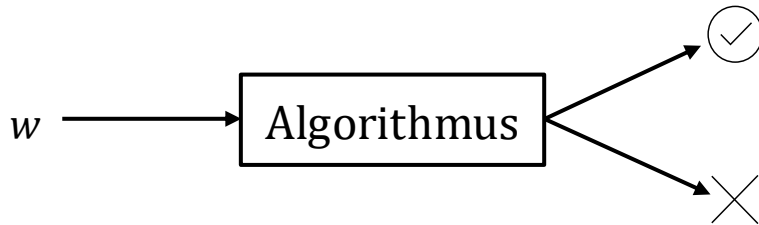
Wortproblem

Sprachen werden in der Literatur oft mit Entscheidungsproblemen identifiziert:

Wortproblem zu A :

Gegeben: Wort $w \in \Sigma^*$

Frage: Ist w ein Element von A ?



Entscheidbar



Semi-Entscheidbar

Zusammenhang zu Turing-Maschinen



Turing-Maschinen sind
zunächst nur Semi-Entscheider!

Können wir TMs etwas
anpassen, dass sie auch
„Entscheider“ werden können?

Statt $(Q, \Sigma, \Gamma, q_0, \delta, Q_F)$ benutze die Form $(Q, \Sigma, \Gamma, q_0, \delta, q_{acc}, q_{rej})$,
also jeweils:

- Ein eindeutiger **akzeptierender Zustand** q_{acc} .
- Ein eindeutiger **abweisender Zustand** q_{rej} .

Diese nennen wir auch **Haltezustände**.

Haltezustände



Diese Definition von TMs ist nicht schwächer!

Diese TMs **halten** in einer akzeptierenden oder abweisenden Konfiguration.

Diese TMs können also:

1. Nach endlich vielen Schritten das Wort akzeptieren.
2. Nach endlich vielen Schritten das Wort abweisen.
3. Unendlich lange laufen, ohne eine Haltekonfiguration zu erreichen – sie loopen.

Bemerkung zu NTMs:

Wir können auch in Zuständen stecken bleiben!
Es reicht aber, wenn es einen Pfad zum akzeptierenden Zustand gibt, um das Wort zu akzeptieren.

Entscheider

Proposition 3.16

Eine Menge $A \subseteq \Sigma^*$ ist genau dann **semi-entscheidbar**, wenn es eine Turing-Maschine M mit $A = \mathcal{L}(M)$ gibt.

Definition 3.17

Wir nennen eine Turing-Maschine $M = (Q, \Sigma, \Gamma, q_0, \delta, q_{acc}, q_{rej})$ **total** oder einen **Entscheider**, wenn jede Berechnung von M zu jeder Eingabe x nach endlich vielen Schritten hält.

Proposition 3.18

Eine Menge $A \subseteq \Sigma^*$ ist genau dann **entscheidbar**, wenn es einen Entscheider M mit $A = \mathcal{L}(M)$ gibt.

Entscheidbare Mengen



Wir sehen später: Das ist ein unentscheidbares Problem!

Bemerkung 3.19

Bereits bekannte Theoreme und Lemma gelten auch für Entscheider:

- Zu jedem nicht-deterministischen Entscheider M existiert ein deterministischer Entscheider M' mit $\mathcal{L}(M) = \mathcal{L}(M')$.
- Zu jedem Mehr-Band-Entscheider M_k existiert ein Ein-Band-Entscheider M' mit $\mathcal{L}(M_k) = \mathcal{L}(M')$.
- Zu jedem Entscheider M_{\leftrightarrow} mit beidseitig unendlichem band existiert ein Entscheider M' mit rechts unendlichem Band und $\mathcal{L}(M_{\leftrightarrow}) = \mathcal{L}(M')$.

Kontextsensitive Sprachen

Lemma 3.20

Jede kontextsensitive Sprache $\mathcal{L}(G)$ ist entscheidbar.

Beweis: Siehe Kapitel 1.

Hilfssatz für entscheidbare Sprachen

Theorem 3.21

Eine Sprache $A \subseteq \Sigma^*$ ist genau dann entscheidbar, wenn A und \bar{A} semi-entscheidbar sind.

“ \Rightarrow ”: Klar.

“ \Leftarrow ”:

Sei M_A eine TM (Semi-Entscheider) für A , und $M_{\bar{A}}$ eine TM für \bar{A} .

Betrachte folgenden Algorithmus:

Eingabe: $w \in \Sigma^*$

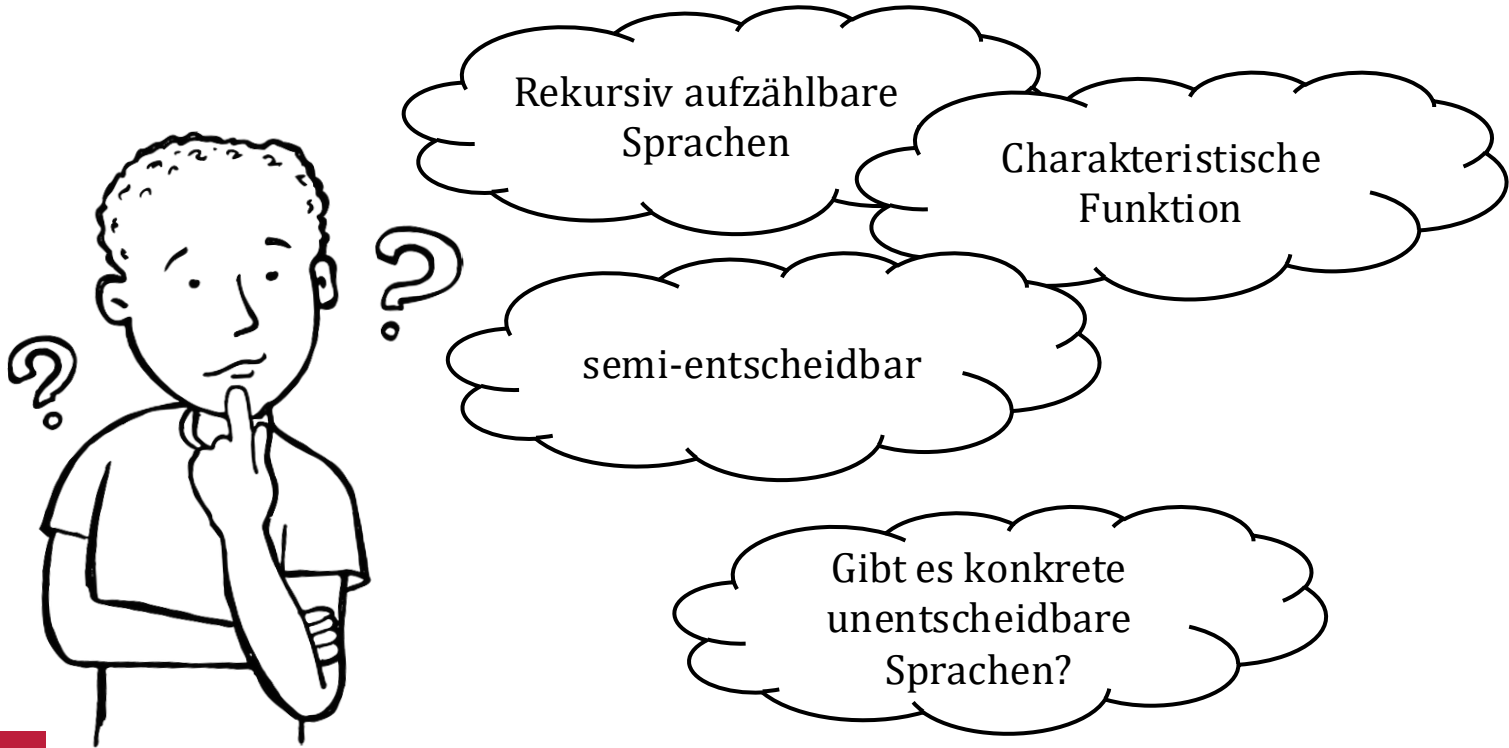
For $i = 1, 2, 3, \dots$

if (M_A akzeptiert Eingabe w in höchstens i Schritten) then return 1

if ($M_{\bar{A}}$ akzeptiert Eingabe w in höchstens i Schritten) then return 0

Dies lässt sich als TM /
Entscheider verpacken!

Zusammenhang der Terminologie



Rekursive Sprachen

Definition 3.22

Eine Sprache $A \subseteq \Sigma^*$ ist **rekursiv aufzählbar**, wenn $A = \emptyset$ gilt, oder es eine totale, berechenbare Funktion $f: \mathbb{N} \rightarrow \Sigma^*$ gibt mit $A = \{f(0), f(1), \dots\} = \{f(i) \mid i \in \mathbb{N}\}$.

Wir sagen, dass A von f rekursiv aufgezählt wird.

Eine Sprache $A \subseteq \Sigma^*$ ist **rekursiv**, falls sowohl A als auch \bar{A} rekursiv aufzählbar sind.

Bemerkung: Beachte, dass $f(i) = f(j)$ für $i \neq j$ erlaubt ist.

Theorem 3.23

Eine Sprache $A \subseteq \Sigma^*$ ist genau dann rekursiv auszählbar, wenn sie semi-entscheidbar ist.

Beweis: Tafel!

Rekursive Sprachen

Korollar 3.24

Eine Sprache $A \subseteq \Sigma^*$ ist genau dann rekursiv, wenn sie entscheidbar ist.

Korollar 3.25

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 $f: \mathbb{N} \rightarrow \Sigma^*$.
5. χ'_A ist berechenbar.
6. A ist der Definitionsbereich einer partiellen berechenbaren Funktion $g: \Sigma^* \rightarrow_p \Sigma_2^*$.

Abzählbar vs. Aufzählbar

Abzählbar

Jedem Element kann eine Zahl aus \mathbb{N} zugewiesen werden.

Beispiele:

1. $\mathbb{N}, \mathbb{Q}, \mathbb{Z}$ sind abzählbar.
2. Jede Sprache $A \subseteq \Sigma^*$ ist abzählbar.
3. Σ^* ist abzählbar.
4. Jede Teilmenge einer abzählbaren Menge ist wieder abzählbar.
5. $\mathbb{R}, \mathcal{P}(\mathbb{N})$ sind nicht abzählbar.
6. Die Menge aller Sprachen ist nicht abzählbar.

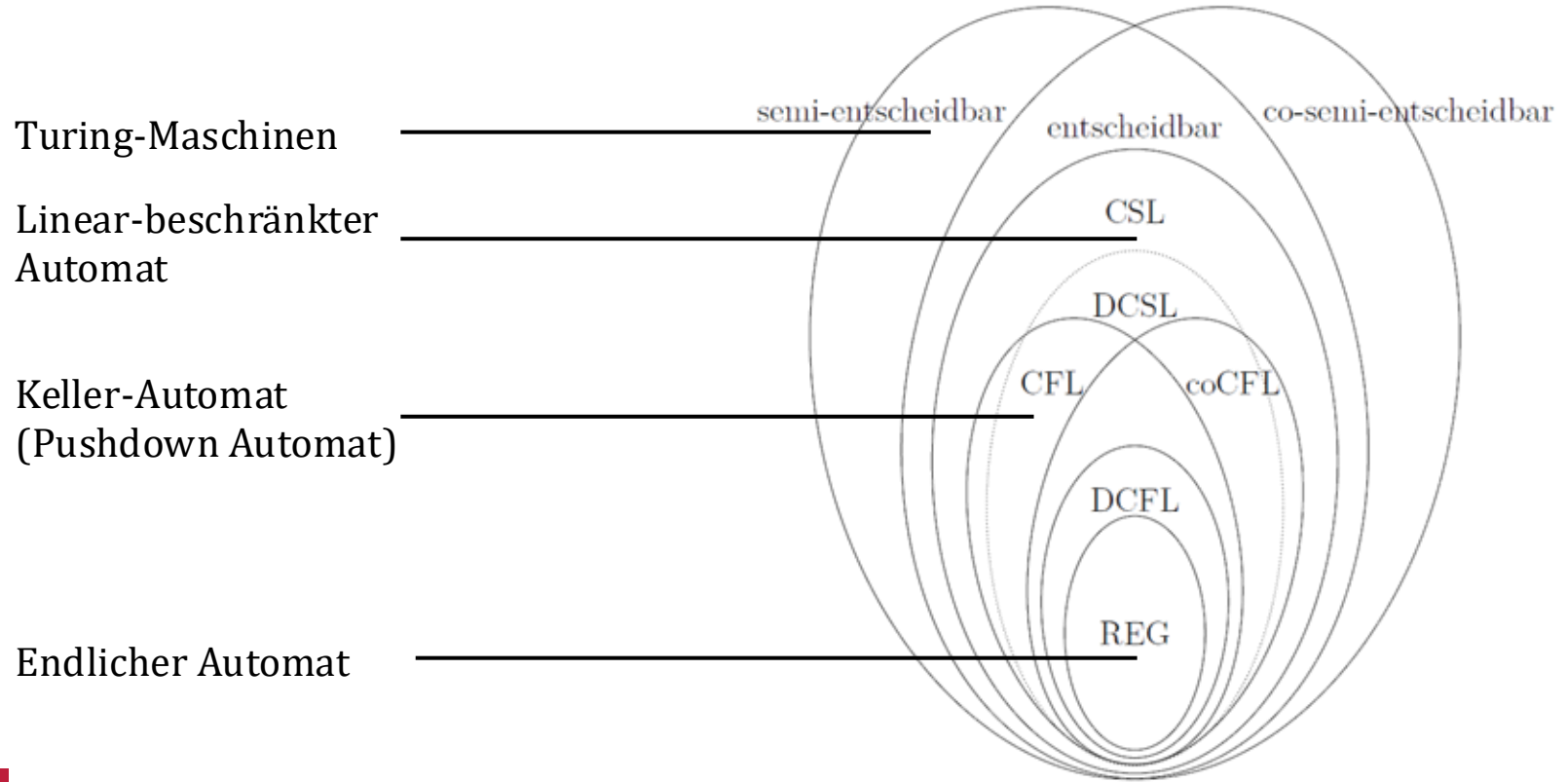
Aufzählbar

Es gibt nicht nur eine Abzählung, sondern die Abzählung muss als Algorithmus implementierbar sein.

Beispiele:

1. Σ^* ist rekursiv aufzählbar.
2. Es existieren Sprachen $A \subseteq \Sigma^*$, die nicht semi-entscheidbar und damit nicht rekursiv aufzählbar sind.

Überblick



Die nächsten Male



Gibt es konkrete
unentscheidbare
Sprachen?

Wie zeige ich
Unentscheidbarkeit?