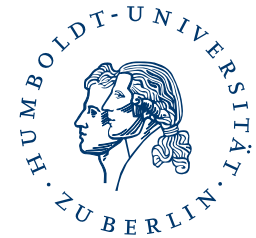
# Einführung in die Theoretische Informatik

Johannes Köbler




Institut für Informatik
Humboldt-Universität zu Berlin

WS 2017/18

# Entscheidbare und semi-entscheidbare Sprachen

## Definition

- Eine NTM $M$ hält bei Eingabe $x$ (kurz: $M(x) = \downarrow$ oder $M(x)\downarrow$), falls alle Rechnungen von $M(x)$ eine endliche Länge haben.
- Falls $M(x)$ nicht hält, schreiben wir auch kurz $M(x) = \uparrow$ oder $M(x)\uparrow$.
- Eine DTM $M$ entscheidet eine Eingabe $x$, falls $M(x)$ hält oder eine Konfiguration mit einem Endzustand erreicht.
- Eine Sprache heißt entscheidbar, falls sie von einer DTM $M$ erkannt wird, die alle Eingaben entscheidet. Die zugehörige Sprachklasse ist

$$\text{REC} = \{L(M) \mid M \text{ ist eine DTM, die alle Eingaben entscheidet}\}$$

- Jede von einer DTM akzeptierte Sprache heißt semi-entscheidbar.

## Bemerkung

- Eine DTM $M$ entscheidet zwar immer alle Eingaben $x \in L(M)$, aber eventuell nicht alle $x \in \overline{L(M)}$. Daher heißt $L(M)$ semi-entscheidbar.
- Später werden wir sehen, dass $\text{RE} = \{L(M) \mid M \text{ ist eine DTM}\}$ ist.

# Berechnung von Funktionen

## Definition

- Eine $k$-DTM $M = (Z, \Sigma, \Gamma, \delta, q_0, E)$ berechnet eine Funktion $f : \Sigma^* \to \Gamma^*$, falls $M$ bei jeder Eingabe $x \in \Sigma^*$ in einer Konfiguration

  $$K = (q, u_1, a_1, v_1, \ldots, u_k, a_k, v_k) \text{ mit } u_k = f(x)$$

  hält (d.h. $K_x \vdash^* K$ und $K$ hat keine Folgekonfiguration).
- Hierfür sagen wir auch, $M$ gibt bei Eingabe $x$ das Wort $f(x)$ aus und schreiben $M(x) = f(x)$.
- $f$ heißt Turing-berechenbar (oder einfach berechenbar), falls es eine $k$-DTM $M$ mit $M(x) = f(x)$ für alle $x \in \Sigma^*$ gibt.
- Aus historischen Gründen werden berechenbare Funktionen auch rekursiv (engl. *recursive*) genannt.

# Berechenbarkeit von charakteristischen Funktionen

## Definition

Für eine Sprache $A \subseteq \Sigma^*$ ist die charakteristische Funktion $\chi_A : \Sigma^* \to \{0, 1\}$ wie folgt definiert:

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

## Bemerkung

- In den Übungen wird gezeigt, dass eine Sprache $A$ genau dann entscheidbar ist, wenn $\chi_A$ berechenbar (also rekursiv) ist.
- Dass CSL echt in REC enthalten ist, wird ebenfalls in den Übungen gezeigt.
- Beispiele für interessante semi-entscheidbare Sprachen, die nicht entscheidbar sind, werden wir noch kennenlernen.
- Somit gilt REG $\subsetneq$ DCFL $\subsetneq$ CFL $\subsetneq$ DCSL $\subseteq$ CSL $\subsetneq$ REC $\subsetneq$ RE.

# Berechenbarkeit von partiellen Funktionen

## Definition

- Eine partielle Funktion hat die Form $f : \Sigma^* \to \Gamma^* \cup \{\uparrow\}$.
- Für $f(x) = \uparrow$ sagen wir auch $f(x)$ ist undefiniert.
- Der Definitionsbereich (engl. *domain*) von $f$ ist

  $$dom(f) = \{x \in \Sigma^* \mid f(x) \neq \uparrow\}.$$

- Das Bild (engl. *image*) von $f$ ist

  $$img(f) = \{f(x) \mid x \in dom(f)\}.$$

- $f$ heißt total, falls $dom(f) = \Sigma^*$ ist.
- Eine partielle Funktion $f$ heißt berechenbar, falls es eine $k$-DTM $M$ mit $M(x) = f(x)$ für alle $x \in \Sigma^*$ gibt (d.h. $M(x)$ gibt für alle $x \in dom(f)$ das Wort $f(x)$ aus und hält im Fall $x \notin dom(f)$ nicht).

# Berechenbarkeit von Funktionen

Wir fassen die (partiellen) berechenbaren Funktionen in folgenden Klassen zusammen:

$$\mathsf{FREC} = \{f \mid f \text{ ist eine berechenbare (totale) Funktion}\},$$

$$\mathsf{FREC}_p = \{f \mid f \text{ ist eine berechenbare partielle Funktion}\}.$$

Dann gilt $\mathsf{FREC} \subsetneq \mathsf{FREC}_p$.

# Berechenbarkeit von Funktionen

## Beispiel

- Bezeichne $x^+$ den lexikografischen Nachfolger von $x \in \Sigma^*$.
- Für $\Sigma = \{0, 1\}$ ergeben sich beispielsweise folgende Werte:

| $x$ | $\varepsilon$ | 0 | 1 | 00 | 01 | 10 | 11 | 000 | $\cdots$ |
|---|---|---|---|---|---|---|---|---|---|
| $x^+$ | 0 | 1 | 00 | 01 | 10 | 11 | 000 | 001 | $\cdots$ |

- Betrachte die auf $\Sigma^*$ definierten partiellen Funktionen $f_1, f_2, f_3, f_4$ mit

$$f_1(x) = 0, \quad f_2(x) = x, \quad f_3(x) = x^+ \qquad \text{und} \qquad f_4(x) = \begin{cases} \uparrow, & x = \varepsilon, \\ y, & x = y^+. \end{cases}$$

- Da $f_1, f_2, f_3, f_4$ berechenbar sind, gehören die totalen Funktionen $f_1, f_2, f_3$ zu FREC und die partielle Funktion $f_4$ zu $\text{FREC}_p$.
- Da $f_4$ keine totale Funktion ist, gehört $f_4$ nicht zu FREC. ◁

# Partielle charakteristische Funktionen

## Definition

Für eine Sprache $A \subseteq \Sigma^*$ ist die partielle charakteristische Funktion $\hat{\chi}_A$ wie folgt definiert:

$$\hat{\chi}_A(x) = \begin{cases} 1, & x \in A \\ \uparrow, & x \notin A \end{cases}$$

## Satz

Eine Sprache $A \subseteq \Sigma^*$ ist genau dann semi-entscheidbar, falls ihre partielle charakteristische Funktion $\hat{\chi}_A$ berechenbar ist.

# Partielle charakteristische Funktionen

**Satz**

Folgende Eigenschaften sind äquivalent:

1. $A$ ist semi-entscheidbar (d.h. $A$ wird von einer DTM akzeptiert),
2. $\hat{\chi}_A$ ist berechenbar (d.h. $\hat{\chi}_A \in \mathrm{FREC}_p$),
3. es gibt eine berechenbare partielle Funktion $f \in \mathrm{FREC}_p$ mit $A = dom(f)$ (d.h. es gibt eine DTM $M$ mit $A = \{x \in \Sigma^* \mid M(x)\downarrow\}$).

**Beweis**

1 $\Rightarrow$ 2 Sei $M$ eine DTM mit $L(M) = A$. Dann lässt sich $M$ so modifizieren, dass sie eine 1 ausgibt (und hält), sobald sie einen Endzustand erreicht, und in eine Endlosschleife geht, sobald sie eine Konfiguration ohne Folgekonfiguration erreicht.

2 $\Rightarrow$ 3 Klar, da der Definitionsbereich $dom(\hat{\chi}_A)$ von $\hat{\chi}_A$ die Sprache $A$ ist.

3 $\Rightarrow$ 1 Sei $M$ eine DTM, die eine partielle Funktion $f$ mit $dom(f) = A$ berechnet. Dann lässt sich $M$ so modifizieren, dass sie genau dann in einen Endzustand übergeht, wenn $M$ eine Konfiguration ohne Folgekonfiguration erreicht. □

# Charakterisierung der rekursiv aufzählbaren Sprachen

## Definition

Eine Sprache $A \subseteq \Sigma^*$ heißt rekursiv aufzählbar, falls $A = \varnothing$ oder das Bild $img(f)$ einer berechenbaren Funktion $f : \Gamma^* \to \Sigma^*$ ist.

## Satz

Folgende Eigenschaften sind äquivalent:

1. $A$ ist semi-entscheidbar (d.h. $A$ wird von einer DTM akzeptiert),
2. $A$ wird von einer 1-DTM akzeptiert,
3. $A$ ist vom Typ 0,
4. $A$ wird von einer NTM akzeptiert,
5. $A$ ist rekursiv aufzählbar.

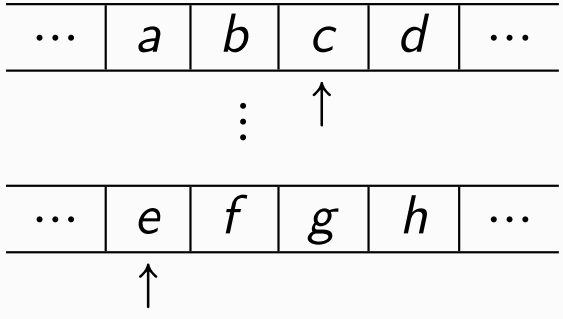
## Beweis

Die Implikationen 2 $\Rightarrow$ 3 $\Rightarrow$ 4 werden in den Übungen gezeigt.
Hier zeigen wir 1 $\Rightarrow$ 2 und 4 $\Rightarrow$ 5 $\Rightarrow$ 1.

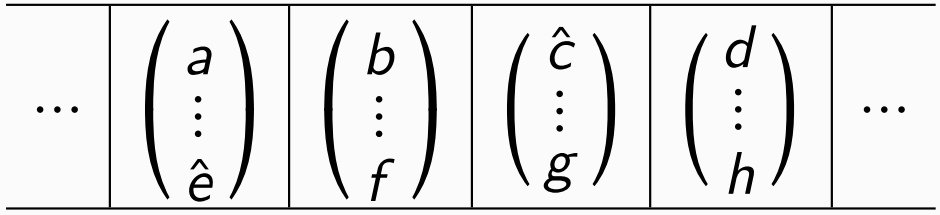
# Simulation einer $k$-DTM durch eine 1-DTM

## Beweis von ❶ $\Rightarrow$ ❷: $\{L(M) \mid M \text{ ist eine DTM}\} \subseteq \{L(M) \mid M \text{ ist eine 1-DTM}\}$

- Sei $M = (Z, \Sigma, \Gamma, \delta, q_0, E)$ eine $k$-DTM mit $L(M) = A$.
- Wir konstruieren eine 1-DTM $M' = (Z', \Sigma, \Gamma', \delta', z_0, E)$ für $A$.
- $M'$ simuliert $M$, indem sie jede Konfiguration $K$ von $M$ der Form




durch eine Konfiguration $K'$ folgender Form nachbildet:



# Simulation einer $k$-DTM durch eine 1-DTM

**Beweis von ❶ $\Rightarrow$ ❷: $\{L(M) \mid M \text{ ist eine DTM}\} \subseteq \{L(M) \mid M \text{ ist eine 1-DTM}\}$**

- Das heißt, $M'$ arbeitet mit dem Alphabet

$$\Gamma' = \Gamma \cup (\Gamma \cup \{\hat{a} \mid a \in \Gamma\})^k$$

- und erzeugt bei Eingabe $x = x_1 \dots x_n \in \Sigma^*$ zuerst die der Startkonfiguration

$$K_x = (q_0, \varepsilon, x_1, x_2 \dots x_n, \varepsilon, \sqcup, \varepsilon, \dots, \varepsilon, \sqcup, \varepsilon)$$

von $M$ bei Eingabe $x$ entsprechende Konfiguration

$$K'_x = q'_0 \begin{pmatrix} \hat{x}_1 \\ \hat{\sqcup} \\ \vdots \\ \hat{\sqcup} \end{pmatrix} \begin{pmatrix} x_2 \\ \sqcup \\ \vdots \\ \sqcup \end{pmatrix} \cdots \begin{pmatrix} x_n \\ \sqcup \\ \vdots \\ \sqcup \end{pmatrix}.$$

# Simulation einer $k$-DTM durch eine 1-DTM

**Beweis von ❶ $\Rightarrow$ ❷: $\{L(M) \mid M \text{ ist eine DTM}\} \subseteq \{L(M) \mid M \text{ ist eine 1-DTM}\}$**

- Dann simuliert $M'$ jeweils einen Schritt von $M$ durch folgende Sequenz von Rechenschritten:
  - Zuerst geht $M'$ solange nach rechts, bis sie alle mit $\hat{}$ markierten Zeichen (z.B. $\hat{a}_1, \ldots, \hat{a}_k$) gefunden hat.
  - Diese Zeichen speichert $M'$ in ihrem Zustand.
  - Anschließend geht $M'$ wieder nach links und realisiert dabei die durch $\delta(q, a_1, \ldots, a_k)$ vorgegebene Anweisung von $M$.
  - Dabei speichert $M'$ den aktuellen Zustand $q$ von $M$ ebenfalls in ihrem Zustand.
- Sobald $M$ in einen Endzustand übergeht, wechselt $M'$ ebenfalls in einen Endzustand und hält.
- Somit gilt $L(M') = L(M)$. □