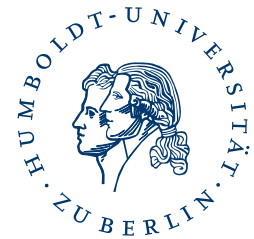
Einführung in die Theoretische Informatik

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

WS 2016/17

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 NTM M entscheidet eine Eingabe x, falls $M(x)$ hält oder eine Konfiguration mit einem Endzustand erreichen kann.
- Eine Sprache $L \subseteq \Sigma^*$ heißt entscheidbar, falls eine DTM M mit $L(M) = L$ existiert, die jede Eingabe $x \in \Sigma^*$ entscheidet.
- Jede von einer DTM M erkannte Sprache heißt semi-entscheidbar.

Bemerkung

- Die von M akzeptierte Sprache $L(M)$ heißt semi-entscheidbar, da M zwar alle Eingaben $x \in L$ entscheidet (aber eventuell nicht alle $x \in \bar{L}$).
- Später werden wir sehen, dass genau die Typ-0 Sprachen semi-entscheidbar sind.

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 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 k-DTM $M = (Z, \Sigma, \Gamma, \delta, q_0, E)$ berechnet f, falls $M(x)$ für alle $x \in dom(f)$ das Wort $f(x)$ ausgibt und für alle $x \notin dom(f)$ keine Ausgabe berechnet (d.h. $M(x) = \uparrow$).

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

$$\mathrm{REC} = \{L(M) \mid M \text{ ist eine DTM, die jede Eingabe entscheidet}\},$$
$$\mathrm{FREC} = \{f \mid f \text{ ist eine berechenbare (totale) Funktion}\},$$
$$\mathrm{FREC}_p = \{f \mid f \text{ ist eine berechenbare partielle Funktion}\}.$$

Dann gilt:

- $\mathrm{FREC} \subsetneq \mathrm{FREC}_p$ und
- $\mathrm{REG} \subsetneq \mathrm{DCFL} \subsetneq \mathrm{CFL} \subsetneq \mathrm{DCSL} \subseteq \mathrm{CSL} \subsetneq \mathrm{REC} \subsetneq \mathrm{RE}$.

Dass CSL echt in REC enthalten ist, wird in den Übungen gezeigt. Beispiele für interessante semi-entscheidbare Sprachen, die nicht entscheidbar sind, werden wir in der Vorlesung kennenlernen.

Berechenbarkeit von partiellen Funktionen

Beispiel

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

x	ε	0	1	00	01	10	11	000	$\cdots$
x^+	0	1	00	01	10	11	000	001	$\cdots$

- Betrachte die auf Σ^* 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 FREC_p.
- Da f_4 keine totale Funktion ist, gehört f_4 nicht zu FREC. ◁

Berechenbarkeit von charakteristischen Funktionen

Definition

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

$$\chi_A(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases} \quad \text{und} \quad \hat{\chi}_A(x) = \begin{cases} 1, & x \in A \\ \uparrow, & x \notin A \end{cases}$$

Satz

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

Berechenbarkeit von charakteristischen 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,
3. A ist Definitionsbereich einer berechenbaren partiellen Funktion f.

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 Der Definitionsbereich von $\hat{\chi}_A$ ist die Sprache A. Folglich hat die partielle Funktion $f = \hat{\chi}_A$ die gewünschte Eigenschaft.

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.

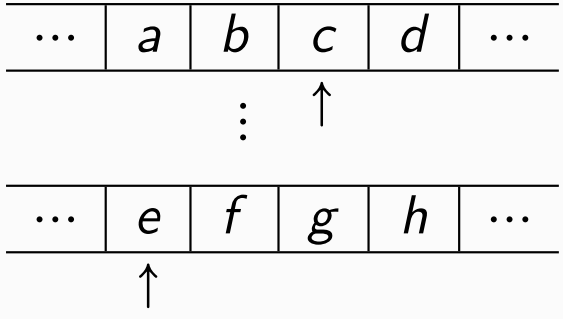
Beweis

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

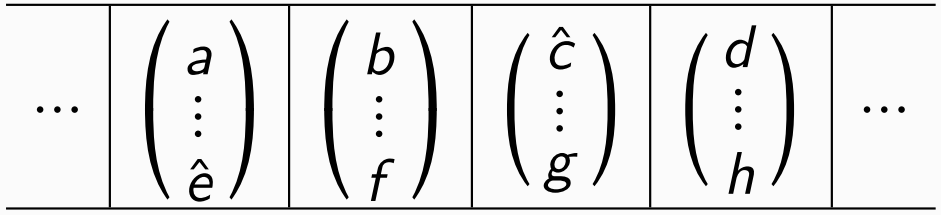
Simulation einer k-DTM durch eine 1-DTM

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.
- Nun ist leicht zu sehen, dass $L(M') = L(M)$ ist. $\square$

Charakterisierung der rekursiv aufzählbaren Sprachen

Beweis von ❹ ⇒ ❺: $\{L(M) \mid M \text{ ist eine NTM}\} \subseteq \{L \mid L \text{ ist rek. aufzählbar}\}$

- Sei $M = (Z, \Sigma, \Gamma, \delta, q_0, E)$ eine k-NTM und sei $A = L(M) \neq \varnothing$.
- Sei $\tilde{\Gamma}$ das Alphabet $Z \cup \Gamma \cup \{\#\}$.
- Wir kodieren eine Konfiguration $K = (q, u_1, a_1, v_1, \ldots, u_k, a_k, v_k)$ durch das Wort

  $$code(K) = \#q\#u_1\#a_1\#v_1\# \ldots \#u_k\#a_k\#v_k\#$$

  und eine Rechnung $K_0 \vdash \cdots \vdash K_t$ durch $code(K_0) \ldots code(K_t)$.
- Dann lassen sich die Wörter von A durch folgende Funktion $f : \tilde{\Gamma}^* \to \Sigma^*$ aufzählen (dabei ist x_0 ein beliebiges Wort in A):

  $$f(x) = \begin{cases} y, & x \text{ kodiert eine akz. Rechnung } K_0 \vdash \cdots \vdash K_t \text{ von} \\ & M(y), \text{ d.h. } K_0 = K_y \text{ und } K_t \in E \times (\Gamma^* \times \Gamma \times \Gamma^*)^k \\ x_0, & \text{sonst} \end{cases}$$

- Da f berechenbar ist, ist $A = img(f)$ rekursiv aufzählbar. □

Charakterisierung der rekursiv aufzählbaren Sprachen

Beweis von 5 $\Rightarrow$ 1: $\{L \mid L \text{ ist rek. aufzählbar}\} \subseteq \{L(M) \mid M \text{ ist eine DTM}\}$

- Sei $f : \Gamma^* \to \Sigma^*$ eine Funktion mit $A = img(f)$ und sei M eine k-DTM, die f berechnet.
- Betrachte folgende $(k+1)$-DTM M', die bei Eingabe x
  - auf dem 2. Band der Reihe nach alle Wörter y in Γ^* erzeugt,
  - für jedes y den Wert $f(y)$ durch Simulation von $M(y)$ berechnet,
  - und ihre Eingabe x akzeptiert, sobald $f(y) = x$ ist. □

Charakterisierung der entscheidbaren Sprachen

Satz

Folgende Eigenschaften sind äquivalent:

1. A ist entscheidbar (d.h. A wird von einer DTM akzeptiert, die alle Eingaben entscheidet),
2. die char. Funktion χ_A von A ist berechenbar,
3. A wird von einer 1-DTM akzeptiert, die bei allen Eingaben hält,
4. A wird von einer NTM akzeptiert, die bei allen Eingaben hält,
5. A und $\bar{A}$ sind vom Typ 0.

Beweis

Die Äquivalenz der Bedingungen 1 bis 4 wird in den Übungen gezeigt. Hier zeigen wir nur die Äquivalenz dieser vier Bedingungen zu 5.

Charakterisierung der entscheidbaren Sprachen

Satz

A ist genau dann entscheidbar, wenn A und $\bar{A}$ semi-entscheidbar sind, d.h. REC = RE $\cap$ co-RE.

Beweis.

- Falls A entscheidbar ist, ist mit χ_A auch $\chi_{\bar{A}}$ berechenbar, d.h. A und $\bar{A}$ sind entscheidbar und damit auch semi-entscheidbar.
- Für die Rückrichtung seien $f_1, f_2 : \Gamma^* \to \Sigma^*$ Turing-berechenbare Funktionen mit $img(f_1) = A$ und $img(f_2) = \bar{A}$.
- Wir betrachten folgende DTM M, die bei Eingabe x für jedes $y \in \Gamma^*$ die beiden Werte $f_1(y)$ und $f_2(y)$ bestimmt und im Fall
  - $f_1(y) = x$ in einem Endzustand hält,
  - $f_2(y) = x$ in einem Nichtendzustand hält.
- Da jede Eingabe x entweder in $img(f_1) = A$ oder in $img(f_2) = \bar{A}$ enthalten ist, hält M bei allen Eingaben, d.h. M entscheidet A. □

Kodierung (Gödelisierung) von Turingmaschinen

- Sei $M = (Z, \Sigma, \Gamma, \delta, q_0, E)$ eine 1-DTM mit
  - Zustandsmenge $Z = \{q_0, \ldots, q_m\}$ (o.B.d.A. sei $E = \{q_m\}$),
  - Eingabealphabet $\Sigma = \{0, 1, \#\}$ und
  - Arbeitsalphabet $\Gamma = \{a_0, \ldots, a_l\}$, wobei wir o.B.d.A. $a_0 = \sqcup$, $a_1 = 0$, $a_2 = 1$ und $a_3 = \#$ annehmen.
- Dann können wir jede Anweisung der Form $q_i a_j \to q_{i'} a_{j'} D$ durch das Wort

  $$\#bin(i)\#bin(j)\#bin(i')\#bin(j')\#b_D\#$$

  kodieren.
- Dabei ist $bin(n)$ die Binärdarstellung von n und

$$b_D = \begin{cases} 0, & D = N, \\ 1, & D = L, \\ 10, & D = R. \end{cases}$$

Kodierung von Turingmaschinen

- M lässt sich nun als ein Wort über dem Alphabet $\{0, 1, \#\}$ kodieren, indem wir die Anweisungen von M in kodierter Form auflisten.
- Kodieren wir die Zeichen $0, 1, \#$ binär (z.B. $0 \mapsto 00$, $1 \mapsto 11$, $\# \mapsto 10$), so gelangen wir zu einer Binärkodierung w_M von M.
- Die durch w_M repräsentierte natürliche Zahl $(w_M)_2$ wird auch die Gödel-Nummer von M genannt.
- M_w ist durch Angabe von w bis auf die Benennung ihrer Zustände und Arbeitszeichen eindeutig bestimmt.
- Ganz analog lassen sich auch k-DTMs mit $k > 1$ (sowie NTMs, Konfigurationen oder Rechnungen von TMs) binär kodieren.
- Umgekehrt können wir jedem Binärstring $w \in \{0, 1\}^*$ eine DTM M_w wie folgt zuordnen:

$$M_w = \begin{cases} M, & \text{falls eine DTM } M \text{ mit } w_M = w \text{ existiert,} \\ M_0, & \text{sonst (dabei sei } M_0 \text{ eine beliebige DTM).} \end{cases}$$