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 $M(x)$ nach endlich vielen Schritten halten.
- 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

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

- 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 $\mathrm{RE} = \{L(M) \mid M \text{ ist eine DTM}\}$ 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 χ_A berechenbar (also rekursiv) ist. Dies erklärt die Bezeichnung REC für die Klasse der entscheidbaren Sprachen.
- Dort wird auch gezeigt, dass CSL echt in REC enthalten ist.
- Beispiele für interessante semi-entscheidbare Sprachen, die nicht entscheidbar sind, werden wir noch kennenlernen.
- Somit gilt $\mathrm{REG} \subsetneq \mathrm{DCFL} \subsetneq \mathrm{CFL} \subsetneq \mathrm{DCSL} \subseteq \mathrm{CSL} \subsetneq \mathrm{REC} \subsetneq \mathrm{RE}$.

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).

Falls M die partielle Fkt. f berechnet, gilt also $dom(f) = \{x \in \Sigma^* \mid M(x)\downarrow\}$. Daher bezeichnen wir diese Menge auch mit $dom(M)$.

Wir fassen die berechenbaren Funktionen und berechenbaren partiellen Funktionen in folgenden Klassen zusammen:

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

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

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

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. ◁

Partielle charakteristische Funktionen

Definition

Sei $A \subseteq \Sigma^*$ eine Sprache.

- Die partielle charakteristische Funktion $\hat{\chi}_A$ von A ist

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

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

Charakterisierung der rekursiv aufzählbaren Sprachen

Satz

Folgende Eigenschaften sind für eine Sprache $A \subseteq \Sigma^*$ 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 rek. aufzählbar (d.h. $A = \varnothing$ oder $A = img(f)$ für eine Fkt. $f \in \text{FREC}$),
6. $\hat{\chi}_A$ ist berechenbar (d.h. $\hat{\chi}_A \in \text{FREC}_p$),
7. es gibt eine DTM M mit $A = dom(M)$.

Beweis

Die Implikationen 2 $\Rightarrow$ 3 $\Rightarrow$ 4 werden in den Übungen gezeigt.
Hier zeigen wir 1 $\Rightarrow$ 2 und 4 $\Rightarrow$ 5 $\Rightarrow$ 6 $\Rightarrow$ 7 $\Rightarrow$ 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

$$
\begin{array}{c|c|c|c|c|c}
\hline
\cdots & a & b & c & d & \cdots \\
\hline
\multicolumn{1}{c}{} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{\vdots} & \multicolumn{1}{c}{\uparrow} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} \\
\hline
\cdots & e & f & g & h & \cdots \\
\hline
\multicolumn{1}{c}{} & \multicolumn{1}{c}{\uparrow} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{} & \multicolumn{1}{c}{}
\end{array}
$$

durch eine Konfiguration K' folgender Form nachbildet:

$$
\begin{array}{c|c|c|c|c|c}
\hline
\cdots & \begin{pmatrix} a \\ \vdots \\ \hat{e} \end{pmatrix} & \begin{pmatrix} b \\ \vdots \\ f \end{pmatrix} & \begin{pmatrix} \hat{c} \\ \vdots \\ g \end{pmatrix} & \begin{pmatrix} d \\ \vdots \\ h \end{pmatrix} & \cdots \\
\hline
\end{array}
$$

Simulation einer k-DTM durch eine 1-DTM

Beweis von ❶ ⇒ 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 ⇒ 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 ˆ 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 $L(M') = L(M)$. □

Charakterisierung der rekursiv aufzählbaren Sprachen

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

- 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):

  $$f(w) = \begin{cases} x, & w \text{ kodiert eine akz. Rechnung } K_0 \vdash \cdots \vdash K_t \text{ von} \\ & M(x) \text{, d.h. } K_0 = K_x \text{ und } K_t \in E \times (\Gamma^* \times \Gamma \times \Gamma^*)^k \\ x_0, & \text{sonst} \end{cases}$$
- Da f berechenbar ist, ist $A = img(f)$ rekursiv aufzählbar. □

Charakterisierung der rekursiv aufzählbaren Sprachen

Beweis von ❺ ⇒ ❻: $\{A \mid A \text{ ist rek. aufzählbar}\} \subseteq \{A \mid \hat{\chi}_A \in \mathsf{FREC}_p\}$

- Sei M eine DTM, die eine Fkt. $f : \Gamma^* \to \Sigma^*$ mit $A = img(f)$ berechnet.
- Dann wird $\hat{\chi}_A$ von der DTM M' berechnet, die bei Eingabe x
  - der Reihe nach für alle $w \in \Gamma^*$ das Wort $f(w)$ berechnet und
  - den Wert 1 ausgibt, sobald $f(w) = x$ ist. □

Beweis von ❻ ⇒ ❼: $\{A \mid \hat{\chi}_A \in \mathsf{FREC}_p\} \subseteq \{dom(M) \mid M \text{ ist eine DTM}\}$

- Sei M eine DTM, die $\hat{\chi}_A$ berechnet.
- Da $dom(\hat{\chi}_A) = A$ ist, folgt $A = dom(M)$. □

Beweis von ❼ ⇒ ❶: $\{dom(M) \mid M \text{ ist eine DTM}\} \subseteq \{L(M) \mid M \text{ ist eine DTM}\}$

- Sei $A = dom(M)$ für eine DTM M.
- Dann gilt $A = L(M')$ für die DTM M', die M simuliert und nur dann in einen Endzustand übergeht, wenn M hält. □

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 charakteristische 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 semi-entscheidbar.

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

Beweis von ❶ ⇒ ❺: REC ⊆ RE ∩ co-RE

- 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.

Beweis von ❺ ⇒ ❶: RE ∩ co-RE ⊆ REC

- Seien M_A und $M_{\bar{A}}$ DTMs, die die partiellen charakteristischen Funktionen $\hat{\chi}_A$ und $\hat{\chi}_{\bar{A}}$ berechnen.
- Betrachte folgende DTM M, die bei Eingabe x für $t = 0, 1, 2, \ldots$ die beiden DTMs M_A und $M_{\bar{A}}$ bei Eingabe x für t Schritte simuliert und
  - in einem Endzustand hält, falls $M_A(x)$ nach t Schritten hält,
  - in einem Nichtendzustand hält, falls $M_{\bar{A}}(x)$ nach t Schritten hält.
- Da jede Eingabe x entweder in $dom(\hat{\chi}_A) = A$ oder in $dom(\hat{\chi}_{\bar{A}}) = \bar{A}$ enthalten ist, hält M bei allen Eingaben.
- Da zudem $L(M) = A$ ist, folgt $A \in \mathrm{REC}$. □

Kodierung (Gödelisierung) von Turingmaschinen

- Um Eigenschaften von TMs algorithmisch untersuchen zu können, müssen wir TMs als Teil der Eingabe kodieren.
- 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. $a_0 = \sqcup$, $a_1 = 0$ und $a_2 = 1$ annehmen.
- Dann können wir eine Anweisung $q_i a_j \to q_{i'} a_{j'} D$ 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. $0 \mapsto 00$, $1 \mapsto 01$, $\# \mapsto 10$), so gelangen wir zu einer Binärkodierung w_M von M.
- Die durch die Binärzahl $w_M = b_n \dots b_0$ repräsentierte natürliche Zahl $(w_M)_2 = \sum_{i=0}^{n} b_i 2^i$ wird auch die Gödel-Nummer von M genannt.
- M_w ist durch Angabe von w_M bzw. $(w_M)_2$ bis auf die Benennung ihrer Zustände und der Arbeitszeichen in $\Gamma \setminus \{\sqcup, 0, 1\}$ 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 (dabei ist M_0 eine beliebige, aber fest gewählte DTM):

$$M_w = \begin{cases} M, & \text{falls eine DTM } M \text{ mit } w_M = w \text{ existiert} \\ M_0, & \text{sonst} \end{cases}$$

Unentscheidbarkeit des Halteproblems

Definition

- Das Halteproblem ist die Sprache

$$H = \left\{ w\#x \;\middle|\; \begin{array}{l} w, x \in \{0,1\}^* \text{ und} \\ \text{die DTM } M_w \text{ hält} \\ \text{bei Eingabe } x \end{array} \right\}$$

- Das spezielle Halteproblem ist

$$K = \left\{ w \in \{0,1\}^* \;\middle|\; \begin{array}{l} \text{die DTM } M_w \\ \text{hält bei Eingabe } w \end{array} \right\}$$

χ_H	w_1	w_2	w_3	$\cdots$
w_1	0	1	0	$\cdots$
w_2	0	1	1	$\cdots$
w_3	1	1	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

χ_K				
w_1	0			
w_2		1		
w_3			0	
$\vdots$				$\ddots$

Satz

$K \in \mathrm{RE} \setminus \mathrm{co\text{-}RE}$.

Semi-Entscheidbarkeit des speziellen Halteproblems

Beweis von $K \in \mathrm{RE}$

- Sei w_h die Kodierung einer DTM, die bei jeder Eingabe (sofort) hält und betrachte die Funktion $f : \{0,1\}^* \to \{0,1\}^*$ mit

$$f(x) = \begin{cases} w, & x \text{ ist die Binärkodierung einer haltenden Rechnung einer DTM } M_w \text{ bei Eingabe } w, \\ w_h, & \text{sonst.} \end{cases}$$

- Da f berechenbar und $img(f) = K$ ist, folgt $K \in \mathrm{RE}$. □

Bemerkung

Ganz ähnlich lässt sich $H \in \mathrm{RE}$ zeigen.

Unentscheidbarkeit des speziellen Halteproblems

Beweisidee

- Sei $B = (b_{ij})$ die durch $b_{ij} = \chi_H(w_i\#w_j) \in \{0,1\}^*$ definierte Binärmatrix.
- Dann kann keine Zeile $b_{i1}b_{i2}\ldots$ von B mit der invertierten Diagonalen $\bar{b}_{11}\bar{b}_{22}\ldots$ von B übereinstimmen, da sonst $b_{ii} = \bar{b}_{ii}$ sein müsste.
- Da aber die i-te Zeile von B wegen

$$b_{ij} = \chi_H(w_i\#w_j) = \chi_{dom(M_{w_i})}(w_j)$$

  die Sprache $dom(M_{w_i}) = \{w_j \in \{0,1\}^* \mid M_{w_i}(w_j)\downarrow\} \in \mathrm{RE}$ kodiert und
- die invertierte Diagonale wegen

$$\bar{b}_{ii} = \chi_{\bar{H}(w_i\#w_i)} = \chi_{\bar{K}}(w_i)$$

  die Sprache $\bar{K}$ kodiert, folgt $\bar{K} \neq dom(M_{w_i})$ für alle $i \geq 1$.
- Dies impliziert $\bar{K} \notin \mathrm{RE}$, da die Zeilen von B wegen

$$\{dom(M_{w_i}) \mid i \geq 1\} = \{A \subseteq \{0,1\}^* \mid A \in \mathrm{RE}\}$$

  alle semi-entscheidbaren Binärsprachen kodieren.

Unentscheidbarkeit des speziellen Halteproblems

Beweis von $\bar{K} \notin \mathrm{RE}$

- Angenommen, die Sprache

$$\bar{K} = \{w \mid M_w(w)\uparrow\} \qquad (*)$$

wäre semi-entscheidbar.

- Dann existiert eine DTM M_{w_i} mit

$$dom(M_{w_i}) = \bar{K} \qquad (**)$$

- Dies führt jedoch auf einen Widerspruch:

$$w_i \in \bar{K} \underset{(*)}{\Leftrightarrow} M_{w_i}(w_i)\uparrow \Leftrightarrow w_i \notin dom(M_{w_i}) \underset{(**)}{\Leftrightarrow} w_i \notin \bar{K} \quad ↯$$

χ_H	w_1	w_2	w_3	$\cdots$
w_1	**0**	1	0	...
w_2	0	**1**	1	...
w_3	1	1	**0**	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
w_i	1	0	1	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	

□

Korollar

$\mathrm{REC} \subsetneq \mathrm{RE}$.

Beweis

Klar, da $K \in \mathrm{RE} - \mathrm{REC}$. □

Der Reduktionsbegriff

Definition

Eine Sprache $A \subseteq \Sigma^*$ heißt auf $B \subseteq \Gamma^*$ reduzierbar (kurz: $A \leq B$), falls eine berechenbare Funktion $f : \Sigma^* \to \Gamma^*$ ex., so dass gilt:

$$\forall x \in \Sigma^* : x \in A \Leftrightarrow f(x) \in B.$$

Beispiel

- Es gilt $K \leq H$ mittels $f : w \mapsto w\#w$, da für alle $w \in \{0,1\}^*$ gilt:

$$w \in K \quad \Leftrightarrow \quad M_w(w)\downarrow \quad \Leftrightarrow \quad w\#w \in H$$

- Es gilt sogar $A \leq H$ für jede Binärsprache $A \in \mathrm{RE}$ mittels $f : x \mapsto w\#x$, wobei w die Kodierung einer DTM M_w mit $dom(M_w) = A$ ist:

$$x \in A \quad \Leftrightarrow \quad M_w(x)\downarrow \quad \Leftrightarrow \quad w\#x \in H$$

$\triangleleft$

Der Vollständigkeitsbegriff

Definition

- Eine Sprache B heißt hart für eine Sprachklasse $\mathcal{C}$ (kurz: $\mathcal{C}$-hart oder $\mathcal{C}$-schwer), falls jede Sprache $A \in \mathcal{C}$ auf B reduzierbar ist:

$$\forall A \in \mathcal{C} : A \leq B.$$

- Eine $\mathcal{C}$-harte Sprache B, die zu $\mathcal{C}$ gehört, heißt $\mathcal{C}$-vollständig.

Beispiel

Das Halteproblem H ist RE-vollständig. Es gilt nämlich

- $H \in \mathrm{RE}$ und
- $\forall A \in \mathrm{RE} : A \leq H$

mittels der Reduktionsfunktion $x \mapsto w\#bin(x)$, wobei M_w eine DTM mit $dom(M_w) = \{bin(x) \mid x \in A\}$ ist. ◁

Bemerkung

Auch das spezielle Halteproblem K ist RE-vollständig (siehe Übungen).

Abschluss von REC unter $\leq$

Definition

Eine Sprachklasse $\mathcal{C}$ heißt unter $\leq$ abgeschlossen, wenn für beliebige Sprachen A, B gilt:

$$A \leq B \wedge B \in \mathcal{C} \Rightarrow A \in \mathcal{C}.$$

Satz

Die Klasse REC ist unter $\leq$ abgeschlossen.

Beweis

- Gelte $A \leq B$ mittels f und sei $B \in \text{REC}$.
- Wegen $B \in \text{REC}$ ex. eine DTM M, die χ_B berechnet.
- Betrachte folgende DTM M':
  - M' berechnet bei Eingabe x zuerst den Wert $f(x)$ und
  - simuliert dann M bei Eingabe $f(x)$.

Abschluss von REC und RE unter $\leq$

Satz

Die Klasse REC ist unter $\leq$ abgeschlossen.

Beweis.

- Gelte $A \leq B$ mittels f und sei $B \in$ REC.
- Dann ex. eine DTM M, die χ_B berechnet.
- Betrachte folgende DTM M':
  - M' berechnet bei Eingabe x zuerst den Wert $f(x)$ und
  - simuliert dann M bei Eingabe $f(x)$.
- Wegen $x \in A \Leftrightarrow f(x) \in B$ ist $\chi_A(x) = \chi_B(f(x))$ und daher folgt

$$M'(x) = M(f(x)) = \chi_B(f(x)) = \chi_A(x).$$

- Also berechnet M' die Funktion χ_A, d.h. $A \in$ REC. $\square$

Bemerkung

Der Abschluss von RE unter $\leq$ folgt analog (siehe Übungen).

H ist nicht entscheidbar

Korollar

- $A \leq B \wedge A \notin \mathrm{REC} \Rightarrow B \notin \mathrm{REC}$,
- $A \leq B \wedge A \notin \mathrm{RE} \Rightarrow B \notin \mathrm{RE}$.

Beweis

Aus der Annahme, dass B entscheidbar (bzw. semi-entscheidbar) ist, folgt wegen $A \leq B$, dass dies auch auf A zutrifft (Widerspruch). $\square$

Bemerkung

Wegen $K \leq H$ überträgt sich somit die Unentscheidbarkeit von K auf H.

Korollar

$H \notin \mathrm{REC}$.

Das Halteproblem bei leerem Band

Definition

Das Halteproblem bei leerem Band ist die Sprache

$$H_0 = \left\{ w \in \{0,1\}^* \,\middle|\, \begin{array}{l} \text{die DTM } M_w \\ \text{hält bei Eingabe } \varepsilon \end{array} \right\}$$

Satz

H_0 ist RE-vollständig.

Beweis

- $H_0 \in \mathrm{RE}$ folgt wegen $H_0 \leq H \in \mathrm{RE}$ mittels der Reduktionsfunktion $w \mapsto w\#\varepsilon$.

χ_H	w_1	w_2	w_3	$\cdots$
w_1	**0**	1	0	$\cdots$
w_2	**0**	1	1	$\cdots$
w_3	**1**	1	0	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

χ_{H_0}	w_1 $(=\varepsilon)$
w_1	**0**
w_2	**0**
w_3	**1**
$\vdots$	$\vdots$

H_0 ist RE-vollständig

Beweis

- $H_0 \in \mathrm{RE}$ folgt wegen $H_0 \le H \in \mathrm{RE}$ mittels der Reduktionsfunktion $w \mapsto w\#\varepsilon$.
- Sei $A \in \mathrm{RE}$ und sei M eine DTM mit $dom(M) = A$.
- Um A auf H_0 zu reduzieren, transformieren wir x in die Kodierung w_x einer DTM M_{w_x}, die zunächst ihre Eingabe durch x ersetzt und dann $M(x)$ simuliert.
- Dann gilt

$$x \in A \iff w_x \in H_0$$

  und somit $A \le H_0$ mittels der Reduktionsfunktion $x \mapsto w_x$.

Korollar

$H_0 \notin \mathrm{REC}$.

Frage

- Kann man einer beliebig vorgegebenen DTM ansehen, ob die von ihr berechnete partielle Funktion eine gewisse Eigenschaft hat?
- Kann man beispielsweise entscheiden, ob eine gegebene DTM bei allen Eingaben hält, also eine totale Funktion berechnet?

Antwort

Nein, außer wenn jede DTM oder keine DTM eine Funktion mit der fraglichen Eigenschaft berechnet.

Bemerkung

Formal lässt sich eine Eigenschaft, die das Ein-/Ausgabeverhalten von DTMs betrifft, durch eine Menge $\mathcal{F}$ von partiellen Wortfunktionen beschreiben. Eine DTM M berechnet dann eine Funktion f mit der Eigenschaft $\mathcal{F}$, wenn $f \in \mathcal{F}$ ist.

Der Satz von Rice

Definition

- Zu einer Klasse $\mathcal{F}$ von partiellen Funktionen definieren wir die Sprache
$$L_{\mathcal{F}} = \left\{ w \in \{0,1\}^* \;\middle|\; \text{die DTM } M_w \text{ ber. eine partielle Funktion in } \mathcal{F} \right\}$$
- Die Eigenschaft $\mathcal{F}$ heißt trivial, wenn $L_{\mathcal{F}} = \emptyset$ oder $L_{\mathcal{F}} = \{0,1\}^*$ ist.

Der Satz von Rice besagt, dass $L_{\mathcal{F}}$ nur für triviale Eigenschaften entscheidbar ist.

Satz (Satz von Rice)

Für jede nicht triviale Eigenschaft $\mathcal{F}$ ist $L_{\mathcal{F}}$ unentscheidbar.

Beispiel

- Betrachte die Sprachen

$$L_1 = \{w \in \{0,1\}^* \mid M_w(0^n) = 0^{n+1} \text{ für alle } n \geq 0\},$$
$$L_2 = \{w \in \{0,1\}^* \mid M_w(x)\uparrow \text{ für alle } x \in \{0,1\}^*\} \text{ und}$$
$$L_3 = \{w \in \{0,1\}^* \mid M_w(x) = \hat{\chi}_K(x) \text{ für alle } x \in \{0,1\}^*\}$$

- Dann gilt $L_i = L_{\mathcal{F}_i}$ für die Eigenschaften

$$\mathcal{F}_1 = \{f \in \mathrm{FREC}_p \mid f(0^n) = 0^{n+1} \text{ für alle } n \geq 0\},$$
$$\mathcal{F}_2 = \{f \in \mathrm{FREC}_p \mid dom(f) \cap \{0,1\}^* = \varnothing\} \text{ und}$$
$$\mathcal{F}_3 = \{f \in \mathrm{FREC}_p \mid f(x) = \hat{\chi}_K(x) \text{ für alle } x \in \{0,1\}^*\}$$

- $\mathcal{F}_1$ ist nicht trivial, da die partiellen Fkten $f, u : \{0,1\}^* \to \{0,1,\uparrow\}^*$ mit $f(x) = x0$ und $u(x) = \uparrow$ berechenbar sind und $f \in \mathcal{F}_1$ sowie $u \notin \mathcal{F}_1$ ist.
- Da zudem $u \in \mathcal{F}_2$ und $\hat{\chi}_K$ sowohl berechenbar als auch in $\mathcal{F}_3$ ist, während $f \notin \mathcal{F}_2$ und $f \notin \mathcal{F}_3$ ist, sind auch $\mathcal{F}_2$ und $\mathcal{F}_3$ nicht trivial.
- Daher sind L_1, L_2 und L_3 nach dem Satz von Rice unentscheidbar.

Beispiel (Fortsetzung)

- Dagegen ist der Satz von Rice nicht auf folgende Sprachen anwendbar:

$$L_4 = \{w \in \{0,1\}^* \mid M_w(x) = \hat{\chi}_{\bar{K}}(x) \text{ für alle } x \in \{0,1\}^*\} \text{ und}$$
$$L_5 = \{w \in \{0,1\}^* \mid M_w(0^n) \text{ hält für alle } n \geq 0 \text{ nach } n \text{ Schritten}\}$$

- Es gilt $L_4 = L_{\mathcal{F}_4}$ für die Eigenschaft $\mathcal{F}_4 = \{\hat{\chi}_{\bar{K}}\}$, d.h. L_4 beschreibt zwar eine semantische Eigenschaft von DTMs, die sich nur auf deren Ein-/Ausgabeverhalten bezieht.
- Da aber $\bar{K} \notin \mathsf{RE}$ und somit $\hat{\chi}_{\bar{K}}$ nicht berechenbar ist, handelt es sich bei $\mathcal{F}_4$ um eine triviale Eigenschaft: $L_4 = L_{\mathcal{F}_4} = \varnothing$.
- Die Sprache L_5 bezieht sich nicht nur auf das Ein-/Ausgabeverhalten von DTMs, sondern auch auf deren Laufzeit.
- Daher existiert für L_5 keine Eigenschaft $\mathcal{F}$ mit $L_5 = L_{\mathcal{F}}$.

Der Satz von Rice

Satz (Satz von Rice)

Für jede nicht triviale Eigenschaft $\mathcal{F}$ ist die Sprache $L_{\mathcal{F}}$ unentscheidbar.

Beweisidee

- Die Idee besteht darin, H_0 auf $L_{\mathcal{F}}$ (oder auf $\overline{L}_{\mathcal{F}}$) zu reduzieren, indem wir für eine gegebene DTM M_w eine DTM $M_{w'}$ konstruieren mit

  $$w \in H_0 \Leftrightarrow M_{w'} \text{ berechnet (k)eine partielle Funktion in } \mathcal{F}.$$

- Hierzu lassen wir $M_{w'}$ bei Eingabe x zunächst einmal die DTM M_w bei Eingabe ε simulieren.
- Falls $w \notin H_0$ ist, berechnet $M_{w'}$ also die überall undefinierte Funktion $u(x) = \uparrow$ für alle $x \in \{0,1\}^*$.
- Damit die Reduktion gelingt, müssen wir nur noch dafür sorgen, dass $M_{w'}$ im Fall $w \in H_0$ eine partielle Funktion f berechnet, die sich bzgl. der Eigenschaft $\mathcal{F}$ von u unterscheidet d.h. $f \in \mathcal{F} \Leftrightarrow u \notin \mathcal{F}$.
- Da $\mathcal{F}$ nicht trivial ist, ex. eine DTM M, die ein solches f berechnet.

Der Satz von Rice

Satz (Satz von Rice)

Für jede nicht triviale Eigenschaft $\mathcal{F}$ ist die Sprache $L_{\mathcal{F}}$ unentscheidbar.

Beweis

- Sei M eine DTM, die eine Funktion f mit $f \in \mathcal{F} \Leftrightarrow u \notin \mathcal{F}$ berechnet.
- Betrachte die Reduktionsfunktion

  $h(w) = w'$, wobei w' die Kodierung einer DTM ist, die bei Eingabe x zunächst die DTM $M_w(\varepsilon)$ simuliert und im Fall, dass $M_w(\varepsilon)$ hält, mit der Simulation von $M(x)$ fortfährt.

- Dann ist $h : w \mapsto w'$ eine totale berechenbare Funktion und es gilt

$$\begin{aligned} w \in H_0 &\quad\Rightarrow\quad M_{w'} \text{ berechnet } f \\ w \notin H_0 &\quad\Rightarrow\quad M_{w'} \text{ berechnet } u. \end{aligned}$$

- Dies zeigt, dass h das Problem H_0 auf $L_{\mathcal{F}}$ (oder auf $\overline{L}_{\mathcal{F}}$) reduziert, und da H_0 unentscheidbar ist, muss auch $L_{\mathcal{F}}$ unentscheidbar sein. □

Der Satz von Rice für Akzeptoren

Der Satz von Rice gilt auch für Eigenschaften, die das Akzeptanzverhalten einer gegebenen Turingmaschine betreffen.

Satz (Satz von Rice für Spracheigenschaften)

Für eine beliebige Sprachklasse $\mathcal{S}$ sei

$$L_{\mathcal{S}} = \{w \in \{0,1\}^* \mid L(M_w) \in \mathcal{S}\}.$$

Dann ist $L_{\mathcal{S}}$ unentscheidbar, außer wenn $L_{\mathcal{S}} = \varnothing$ oder $L_{\mathcal{S}} = \{0,1\}^*$ ist.

Beweis

Siehe Übungen.

Entscheidungsprobleme für Sprachklassen

Neben dem Wortproblem sind für eine Sprachklasse $\mathcal{C}$ auch folgende Entscheidungsprobleme interessant:

Das Leerheitsproblem ($\mathrm{LP}_{\mathcal{C}}$)

Gegeben: Eine Sprache L aus $\mathcal{C}$.

Gefragt: Ist $L = \varnothing$?

Das Äquivalenzproblem ($\mathrm{\ddot{A}P}_{\mathcal{C}}$)

Gegeben: Zwei Sprachen L_1 und L_2 aus $\mathcal{C}$.

Gefragt: Gilt $L_1 = L_2$?

Das Schnittproblem ($\mathrm{SP}_{\mathcal{C}}$)

Gegeben: Zwei Sprachen L_1 und L_2 aus $\mathcal{C}$.

Gefragt: Ist $L_1 \cap L_2 = \varnothing$?

Hierbei repräsentieren wir Sprachen in $\mathcal{C} = \mathsf{REG}, \mathsf{CFL}, \mathsf{CSL}, \mathsf{RE}$ durch entsprechende Grammatiken und Sprachen in $\mathcal{C} = \mathsf{DCFL}, \mathsf{DCSL}$ durch entsprechende Akzeptoren (also DPDAs bzw. DLBAs).

Das Postsche Korrespondenzproblem (PCP)

Definition

- Sei Σ ein beliebiges Alphabet mit $\# \notin \Sigma$.
- Das Postsche Korrespondenzproblem über Σ (kurz PCP_Σ) ist:

  gegeben: k Wortpaare $(x_1, y_1), \dots, (x_k, y_k) \in \Sigma^+ \times \Sigma^*$

  gefragt: Gibt es eine Folge $\alpha = (i_1, \dots, i_n)$, $n \geq 1$, von Indizes $i_j \in \{1, \dots, k\}$ mit $x_{i_1} \dots x_{i_n} = y_{i_1} \dots y_{i_n}$?
- Das modifizierte PCP über Σ (kurz MPCP_Σ) fragt nach einer Lösung $\alpha = (i_1, \dots, i_n)$ mit $i_1 = 1$.
- Wir notieren eine PCP-Instanz meist in Form einer Matrix $\binom{x_1 \dots x_k}{y_1 \dots y_k}$ und kodieren sie durch das Wort $x_1 \# y_1 \# \dots \# x_k \# y_k$.

Beispiel

Die Instanz $I = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{pmatrix} = \begin{pmatrix} a & ab & caa \\ aca & bc & aa \end{pmatrix}$ besitzt wegen

$$x_1 x_3 x_2 x_3 = acaaabcaa$$

$$y_1 y_3 y_2 y_3 = acaaabcaa$$

die PCP-Lösung $\alpha = (1, 3, 2, 3)$, die auch eine MPCP-Lösung ist. $\triangleleft$

Das Postsche Korrespondenzproblem

Lemma

Für jedes Alphabet Σ gilt $\mathrm{PCP}_\Sigma \leq \mathrm{PCP}_{\{0,1\}}$.

Beweis

- Sei $\Sigma = \{a_1, \ldots, a_m\}$ und sei $k = \max(1, \lceil \log_2(m) \rceil)$. Dann können wir a_i durch eine k-stellige Binärzahl $bin_k(a_i)$ mit dem Wert $i - 1$ und ein Wort $w = w_1 \ldots w_n$ durch $bin(w) = bin_k(w_1) \ldots bin_k(w_n)$ kodieren.
- Nun folgt $\mathrm{PCP}_\Sigma \leq \mathrm{PCP}_{\{0,1\}}$ mittels der Reduktionsfunktion

$$f : \binom{x_1 \ldots x_k}{y_1 \ldots y_k} \mapsto \binom{bin(x_1) \ldots bin(x_k)}{bin(y_1) \ldots bin(y_k)}.$$

□

Beispiel

Sei $\Sigma = \{a, b, c\}$. Dann ist $k = \max(1, \lceil \log_2(3) \rceil) = 2$ und $bin_2(a) = 00$, $bin_2(b) = 01$ und $bin_2(c) = 10$. Somit ist

$$f\begin{pmatrix} a & ab & caa \\ aca & bc & aa \end{pmatrix} = \begin{pmatrix} 00 & 0001 & 100000 \\ 001000 & 0110 & 0000 \end{pmatrix}$$

Das Postsche Korrespondenzproblem

Wir schreiben für $\mathrm{PCP}_{\{0,1\}}$ auch PCP (bzw. MPCP für $\mathrm{MPCP}_{\{0,1\}}$).

Satz

$\mathrm{MPCP} \le \mathrm{PCP}$.

Beweis

- Wir zeigen $\mathrm{MPCP} \le \mathrm{PCP}_\Sigma$ für $\Sigma = \{0, 1, \langle, |, \rangle\}$.
- Für ein Wort $w = w_1 \dots w_n$ sei

$\overset{\longmapsfrom}{w}$	$\overleftarrow{w}$	$\overset{\vdash}{w}$	$\overset{\dashv}{w}$
$\langle w_1 \vert \dots \vert w_n \vert$	$\langle w_1 \vert \dots \vert w_n$	$\vert w_1 \vert \dots \vert w_n$	$w_1 \vert \dots \vert w_n \vert$

- Wir reduzieren MPCP mittels folgender Funktion f auf PCP_Σ:

$$f : \begin{pmatrix} x_1 & \dots & x_k \\ y_1 & \dots & y_k \end{pmatrix} \mapsto \begin{pmatrix} \overset{\longmapsfrom}{x_1} & \overset{\dashv}{x_1} & \dots & \overset{\dashv}{x_k} & \rangle \\ \overleftarrow{y_1} & \overset{\vdash}{y_1} & \dots & \overset{\vdash}{y_k} & |\rangle \end{pmatrix}$$

Das Postsche Korrespondenzproblem

Beweis

- Wir zeigen $\mathrm{MPCP} \leq \mathrm{PCP}_\Sigma$ für $\Sigma = \{0, 1, \langle, |, \rangle\}$.
- Für ein Wort $w = w_1 \dots w_n$ sei

$\overset{\leftarrow\!\mid}{w}$	$\overleftarrow{w}$	$\overset{\mid\!-}{w}$	$\overset{-\!\mid}{w}$
$\langle w_1 \| \dots \| w_n \|$	$\langle w_1 \| \dots \| w_n$	$\| w_1 \| \dots \| w_n$	$w_1 \| \dots \| w_n \|$

- Wir reduzieren MPCP mittels folgender Funktion f auf PCP_Σ:

$$f : \begin{pmatrix} x_1 & \dots & x_k \\ y_1 & \dots & y_k \end{pmatrix} \mapsto \begin{pmatrix} \overset{\leftarrow\!\mid}{x_1} & \overset{-\!\mid}{x_1} & \dots & \overset{-\!\mid}{x_k} & \rangle \\ \overleftarrow{y_1} & \overset{\mid\!-}{y_1} & \dots & \overset{\mid\!-}{y_k} & |\rangle \end{pmatrix}$$

Beispiel

$$f : \begin{pmatrix} 00 & 1 & 101 & 11 \\ 001 & 11 & 0 & 1 \end{pmatrix} \mapsto \begin{pmatrix} \langle|0|0| & 0|0| & 1| & 1|0|1| & 1|1| & \rangle \\ \langle|0|0|1 & |0|0|1 & |1|1 & |0 & |1 & |\rangle \end{pmatrix}$$

◁

Beweis von $\mathrm{MPCP} \le \mathrm{PCP}$

- Wir reduzieren MPCP mittels folgender Funktion f auf PCP_Σ:

$$f : \begin{pmatrix} x_1 & \dots & x_k \\ y_1 & \dots & y_k \end{pmatrix} \mapsto \begin{pmatrix} \overleftrightarrow{x_1} & \overrightarrow{x_1} & \dots & \overrightarrow{x_k} & \rangle \\ \overleftarrow{y_1} & \overleftarrow{y_1} & \dots & \overleftarrow{y_k} & |\rangle \end{pmatrix}$$

- Da jede MPCP-Lösung $\alpha = (1, i_2, \dots, i_n)$ für I auf eine PCP-Lösung $\alpha' = (1, i_2 + 1, \dots, i_n + 1, k + 2)$ für $f(I)$ führt, folgt

$$I \in \mathrm{MPCP} \Rightarrow f(I) \in \mathrm{PCP}_\Sigma.$$

- Für die umgekehrte Implikation sei $\alpha' = (i_1, \dots, i_n)$ eine PCP-Lösung für $f(I)$.
- Dann muss $i_1 = 1$ sein, da nur $\overleftrightarrow{x_1}$ und $\overleftarrow{y_1}$ mit dem gleichen Zeichen beginnen. Zudem muss $i_n = k + 2$ sein, da nur $\rangle$ und $|\rangle$ mit dem gleichen Zeichen enden.
- Wählen wir α' von minimaler Länge, so ist $i_j \in \{2, \dots, k + 1\}$ für $j = 2, \dots, n - 1$.
- Folglich ist $\alpha = (i_1, i_2 - 1, \dots, i_{n-1} - 1)$ eine MPCP-Lösung für I. $\square$

Unentscheidbarkeit des PCP

Satz

PCP ist RE-vollständig und damit unentscheidbar.

Beweis.

- PCP ist semi-entscheidbar, da eine DTM systematisch nach einer Lösung suchen kann.
- Um zu zeigen, dass PCP RE-hart ist, sei A eine beliebige Sprache in RE und sei $G = (V, \Sigma, P, S)$ eine Typ-0 Grammatik für A.
- Wir zeigen $A \le \mathrm{MPCP}_\Gamma$ für $\Gamma = V \cup \Sigma \cup \{\langle, |, \rangle\}$.
- Wegen $\mathrm{MPCP}_\Gamma \le \mathrm{PCP}$ folgt hieraus $A \le \mathrm{PCP}$.

Beweisidee für die Reduktion $A \le \mathrm{MPCP}_\Gamma$:

Transformiere eine Eingabe $w \in \Sigma^*$ in eine Instanz $f(w) = \binom{x_1 \ldots x_k}{y_1 \ldots y_k}$, so dass $\alpha = (i_1, \ldots, i_n)$ genau dann eine MPCP-Lösung für $f(w)$ ist, wenn das zugehörige Lösungswort $x_{i_1} \ldots x_{i_n} = y_{i_1} \ldots y_{i_n}$ eine Ableitung $S = \alpha_0 \Rightarrow \cdots \Rightarrow \alpha_m = w$ von w kodiert.

Beweis von $A \leq \mathrm{MPCP}_\Gamma$

- Wir bilden $f(w)$ aus folgenden Wortpaaren:
  - $(\langle\,, \langle\,|\,S)$, „Startpaar“
  - für jede Regel $l \to r$ in P: (l, r), „Ableitungspaare“
  - für alle $a \in V \cup \Sigma \cup \{|\}$: (a, a), „Kopierpaare“
  - sowie das Paar $(w\,|\,\rangle, \rangle)$ „Abschlusspaar“

Unentscheidbarkeit des PCP

Beispiel

- Sei $G = (\{S\}, \{a, b\}, \{S \to aSbS, \varepsilon\}, S)$ und $w = aabb$.
- Die MPCP-Instanz $f(aabb)$ enthält dann die acht Wortpaare

$$f(aabb) = \begin{pmatrix} \langle & S & S & S & a & b & | & aabb|\rangle \\ \langle|S & aSbS & \varepsilon & S & a & b & | & \rangle \end{pmatrix}.$$

- Der Ableitung $\underline{S} \Rightarrow a\underline{S}bS \Rightarrow aaSb\underline{S}bS \Rightarrow aa\underline{S}bbS \Rightarrow aabb\underline{S} \Rightarrow aabb$ entspricht dann das MPCP-Lösungswort

$$\begin{matrix} \langle|S|aSbS|aaSbSbS|aaSbbS|aabbS|aabb|\rangle \\ \langle|S|aSbS|aaSbSbS|aaSbbS|aabbS|aabb|\rangle \end{matrix}$$

- Das kürzeste MPCP-Lösungswort für $f(aabb)$ ist

$$\begin{matrix} \langle|S|aSbS|aaSbSb|aabb|\rangle \\ \langle|S|aSbS|aaSbSb|aabb|\rangle \end{matrix}$$

- Dieses entspricht der „parallelisierten" Ableitung

$$\underline{S} \Rightarrow a\underline{S}b\underline{S} \Rightarrow^2 aa\underline{S}b\underline{S}b \Rightarrow^2 aabb$$

◁

Unentscheidbarkeit des PCP

Beweis von $A \leq \mathrm{MPCP}_\Gamma$

- Wir bilden $f(w)$ aus folgenden Wortpaaren:
  - $(\langle, \langle | S)$, „Startpaar“
  - für jede Regel $l \to r$ in P: (l, r), „Ableitungspaare“
  - für alle $a \in V \cup \Sigma \cup \{|\}$: (a, a), „Kopierpaare“
  - sowie das Paar $(w|\rangle, \rangle)$ „Abschlusspaar“
- Nun lässt sich leicht aus einer Ableitung $S = \alpha_0 \Rightarrow \cdots \Rightarrow \alpha_m = w$ von w in G eine MPCP-Lösung mit dem Lösungswort

  $$\langle | \alpha_0 | \alpha_1 | \ldots | \alpha_m | \rangle$$

  angeben.
- Umgekehrt lässt sich aus jeder MPCP-Lösung auch eine Ableitung von w in G gewinnen, womit

  $$w \in L(M) \Leftrightarrow f(w) \in \mathrm{MPCP}_\Gamma$$

  gezeigt ist. $\square$

Unentscheidbarkeit des Schnittproblems für CFL

Das Schnittproblem für kontextfreie Grammatiken ($\mathrm{SP}_{\mathsf{CFL}}$)

Gegeben: Zwei kontextfreie Grammatiken G_1 und G_2.

Gefragt: Ist $L(G_1) \cap L(G_2) \neq \emptyset$?

Satz

Das Schnittproblem für kontextfreie Grammatiken ist RE-vollständig.

Unentscheidbarkeit des Schnittproblems für CFL

Satz

Das Schnittproblem für kontextfreie Grammatiken ist RE-vollständig.

Beweis

- Das Problem $\mathrm{SP}_{\mathsf{CFL}}$ ist semi-entscheidbar, da eine DTM systematisch nach einem Wort $x \in L(G_1) \cap L(G_2)$ suchen kann.
- Um PCP auf $\mathrm{SP}_{\mathsf{CFL}}$ zu reduzieren, betrachten wir für eine Folge $s = (x_1, \dots, x_k)$ von Strings $x_i \in \{0,1\}^*$ die Sprache

  $$L_s = \{i_n \dots i_1 \# x_{i_1} \cdots x_{i_n} \mid n \geq 1, 1 \leq i_1, \dots, i_n \leq k\}$$

  über dem Alphabet $\Sigma = \{1, \dots, k, \#, 0, 1\}$.
- Die Sprache L_s wird von der Grammatik $G_s = (\{A\}, \Sigma, P_s, A)$ mit der Regelmenge

  $$P_s\colon A \to 1Ax_1, \dots, kAx_k, 1\#x_1, \dots, k\#x_k$$

  erzeugt.