

Funktion $f: \Sigma^* \rightarrow \Gamma^*$ berechenbar :

Es gibt Turing Maschine M , die für jedes $x \in \Sigma^*$ bei Eingabe x auf dem Eingabeband mit Ausgabe $f(x) \in \Gamma^*$ auf dem Ausgabeband stoppt.

Funktion $f: \Sigma^* \rightarrow \Gamma^*$ berechenbar :

Es gibt Turing Maschine M , die für jedes $x \in \Sigma^*$ bei Eingabe x auf dem Eingabeband mit Ausgabe $f(x) \in \Gamma^*$ auf dem Ausgabeband stoppt.

Funktion $f: A \rightarrow B$ berechenbar :

Es gibt Turing Maschine M , die für jedes $a \in A$ bei Eingabe $\text{cod}(a)$ auf dem Eingabeband mit Ausgabe $\text{cod}(f(a))$ auf dem Ausgabeband stoppt.

Die folgenden sind äquivalent für $A \subseteq \Sigma^*$:

- A ist Turing akzeptierbar
- Es gibt TM M mit $L(M) = A$
- A ist semi-entscheidbar
- χ_A^+ ist berechenbar
- A ist rekursiv aufzählbar
- A wird von Typ-0 Grammatik generiert
- A ist Definitionsbereich einer berechenbaren Funktion
- A ist Wertebereich einer berechenbaren Funktion

$A \subseteq \Sigma^*$ heißt **semi-entscheidbar**, wenn die positive charakteristische Funktion von A berechenbar ist.

$$\chi_A^+ : \Sigma^* \rightarrow \{0,1\} : \chi_A^+(x) = \begin{cases} 1 & x \in A \\ \text{undef} & x \notin A \end{cases}$$

Die folgenden sind äquivalent für $A \subseteq \Sigma^*$:

- A ist Turing akzeptierbar
- Es gibt TM M mit $L(M) = A$
- A ist semi-entscheidbar
- χ_A^+ ist berechenbar
- A ist rekursiv aufzählbar
- A wird von Typ-0 Grammatik generiert
- A ist Definitionsbereich einer berechenbaren Funktion
- A ist Wertebereich einer berechenbaren Funktion

$A \subseteq \Sigma^*$ heißt **rekursiv aufzählbar**, wenn es eine überall berechenbare Funktion $F: \mathbb{N} \rightarrow \Sigma^*$ gibt, sodass

$$A = \{ F(0), F(1), F(2), \dots \} = \{ F(n) \mid n \in \mathbb{N} \}$$

Die Funktion F "zählt die Menge A auf"

- A ist Turing akzeptierbar
- Es gibt TM M mit $L(M) = A$
- A ist semi-entscheidbar
- χ_A^+ ist berechenbar
- A ist rekursiv aufzählbar
- A wird von Typ-0 Grammatik generiert
- A ist Definitionsbereich einer berechenbaren Funktion
- A ist Wertebereich einer berechenbaren Funktion

$$\chi_A : \Sigma^* \rightarrow \{0,1\} : \chi_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$
$$\chi_A : \Sigma^* \rightarrow \{0,1\} : \chi_A(x) = \begin{cases} 1 & x \in A \\ 0 & x \notin A \end{cases}$$

A ist entscheidbar $\Leftrightarrow$ A ist semi-entscheidbar und $\overline{A}$ ist semi-entscheidbar

Σ	Eingabealphabet				
$\Gamma \supset \Sigma$	Bandalphabet				
$\# \in \Gamma \setminus \Sigma$	Leerzeichen (Inhalt von nicht initialisierten Zellen)				
Q	Zustandsmenge (endlich)				
$s \in Q$	Startzustand				
$F \subset Q$	Endzustände				
$\Delta \subset (Q \times \Gamma) \times (Q \times \Gamma \times \{L, B, R\})$	Rechenschrittregeln				
derzeitiger Zustand	derzeitiges gelesenes Symbol	neuer Zustand	geschriebenes Symbol	Kopfbewegung L = links R = rechts B = bleibe	

$$\begin{aligned}\Sigma &= \{a_1, \dots, a_i\} \\ \Gamma &= \{a_1, \dots, a_i, \dots, a_g\} \cup \{\#\} \\ Q &= \{q_1, \dots, q_m\} \\ s &= q_1 \\ F &= \{q_2\} \\ \Delta \subset (Q \times \Gamma) \times (Q \times \Gamma \times \{L, B, R\}) \quad L = \mu_1, B = \mu_2, R = \mu_3 \\ \Delta &= \{\delta_1, \dots, \delta_n\}\end{aligned}$$
$$\text{cod}(M) = \text{cod}(\delta_1) \text{cod}(\delta_2) \cdots \text{cod}(\delta_p)$$

Kodierung von Turing Maschinen als binäre Strings

$$M = (\Sigma, \Gamma, \#, Q, s, F, \Delta)$$

o.B.d.A

$$\Sigma = \{ a_1, \dots, a_t \}$$

$$\Gamma = \{ a_1, \dots, a_t, \dots, a_g \} \cup \{ \# \}$$

$$Q = \{ q_1, \dots, q_m \}$$

$$s = q_1$$

$$F = \{ q_2 \}$$

$$\Delta \subset (Q \times \Gamma) \times (Q \times \Gamma \times \{L, B, R\}) \quad L = \mu_1, B = \mu_2, R = \mu_3$$

$$\Delta = \{ \delta_1, \dots, \delta_D \}$$

$$\text{cod}(M) = \text{cod}(\delta_1) \text{cod}(\delta_2) \dots \text{cod}(\delta_D)$$

$$\text{cod}(q_i, a_j, q_k, \mu_p) = 0 \ 1^h \ 0 \ 1^i \ 0 \ 1^j \ 0 \ 1^k \ 0 \ 1^p \ 0$$

Dekodieren: binäre Strings als Turing Maschinen interpretieren

$$w \in \{0,1\}^*$$

$$M_w = \begin{cases} M \text{ mit } \text{cod}(M)=w & \text{falls } w \text{ eine legale TM Kodierung} \\ \text{TM die nichts akzeptiert} & \text{sonst} \end{cases}$$

Dekodieren: binäre Strings als Turing Maschinen interpretieren

$$w \in \{0,1\}^*$$

$$M_w = \begin{cases} M \text{ mit } \text{cod}(M)=w & \text{falls } w \text{ eine legale TM Kodierung} \\ \text{TM die nichts akzeptiert} & \text{sonst} \end{cases}$$

$$\text{Es gilt: } M_{\text{cod}(M)} \approx M$$

Kodieren beliebiger Strings als binäre Strings

$$\Sigma = \{ a_1, \dots, a_t \}$$

$$x \in \Sigma^* \text{ mit } x = a_{i_1} a_{i_2} a_{i_3} \dots a_{i_n}$$

$$\text{cod}(x) = 0^{i_1} 1^{i_2} 0^{i_3} 1^{i_4} \dots$$

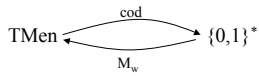
Dekodieren: Interpretation binärer Strings als allgemeine Strings

$$\text{decod}(0^{i_1} 1^{i_2} 0^{i_3} 1^{i_4} \dots) = a_{i_1} a_{i_2} a_{i_3} a_{i_4} \dots$$

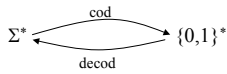
$$\text{decod}(1^{i_1} 0^{i_2} 1^{i_3} 0^{i_4} \dots) = a_{i_1} a_{i_2} a_{i_3} a_{i_4} \dots$$

$$\text{Es gilt: } \text{decod}(\text{cod}(x)) = x$$

Kodieren/Dekodieren von Turing Maschinen und Strings



$$M_{\text{cod}(M)} \approx M$$



$$\text{decod}(\text{cod}(x)) = x$$

Selbst-Akzeptierende Maschinen

$$\text{SAM} = \{ w \in \{0,1\}^* \mid w \in L(M_w) \}$$

$$\overline{\text{SAM}} = \{ w \in \{0,1\}^* \mid w \notin L(M_w) \}$$

Satz: $\overline{\text{SAM}} = \{ w \in \{0,1\}^* \mid w \notin L(M_w) \}$
ist **keine** TM-Sprache (keine semi-entscheidbare Sprache,
keine rekursiv aufzählbare Sprache).

Satz: $\overline{\text{SAM}} = \{ w \in \{0,1\}^* \mid w \notin L(M_w) \}$
ist **keine** TM-Sprache (keine semi-entscheidbare Sprache,
keine rekursiv aufzählbare Sprache).

Beweis: Müssen zeigen:

Es gibt **keine** TM M mit $L(M) = \overline{\text{SAM}}$.

Satz: $\overline{\text{SAM}} = \{ w \in \{0,1\}^* \mid w \notin L(M_w) \}$
ist **keine** TM-Sprache (keine semi-entscheidbare Sprache,
keine rekursiv aufzählbare Sprache).

Beweis: Müssen zeigen:

Es gibt **keine** TM M mit $L(M) = \overline{\text{SAM}}$.

Für **jede** TM M gilt $L(M) \neq \overline{\text{SAM}}$.

Satz: $\overline{\text{SAM}} = \{ w \in \{0,1\}^* \mid w \notin L(M_w) \}$
ist **keine** TM-Sprache (keine semi-entscheidbare Sprache,
keine rekursiv aufzählbare Sprache).

Beweis: Müssen zeigen:

Es gibt **keine** TM M mit $L(M) = \overline{\text{SAM}}$.

Für **jede** TM M gilt $L(M) \neq \overline{\text{SAM}}$.

Für **jedes** $w \in \{0,1\}^*$ gilt $L(M_w) \neq \overline{\text{SAM}}$.

Satz: $\overline{SAM} = \{ w \in \{0,1\}^* \mid w \notin L(M_w) \}$
 ist **keine** TM-Sprache (keine semi-entscheidbare Sprache,
 keine rekursiv aufzählbare Sprache).

Beweis: Müssen zeigen:

Es gibt **keine** TM M mit $L(M) = \overline{SAM}$.

Für **jede** TM M gilt $L(M) \neq \overline{SAM}$.

Für **jedes** $w \in \{0,1\}^*$ gilt $L(M_w) \neq \overline{SAM}$.

$L(M_w)$ und $\overline{SAM}$ unterscheiden sich in w !!

$w \notin L(M_w) \Leftrightarrow w \in \overline{SAM}$

Satz: $SAM = \{ w \in \{0,1\}^* \mid w \in L(M_w) \}$
 ist **nicht** entscheidbar.

Satz: $SAM = \{ w \in \{0,1\}^* \mid w \in L(M_w) \}$
 ist **nicht** entscheidbar.

Beweis:

Wäre SAM entscheidbar, dann müsste

SAM TM akzeptierbar und auch

$\overline{SAM}$ TM akzeptierbar sein.

Widerspruch.

Analog: „spezielles Halteproblem“

Ob eine SML Funktion mit
 seinem eigenen Programmtext als Argument
 terminiert oder nicht,
 ist im Allgemeinen **nicht entscheidbar**.

Analog: „spezielles Halteproblem“

Ob eine SML Funktion mit
 seinem eigenen Programmtext als Argument
 terminiert oder nicht,
 ist im Allgemeinen **nicht entscheidbar**.

Das heißt, es gibt kein automatisches Werkzeug,
 das aus dem Programmtext einer SML Funktion schließen
 kann, ob diese Funktion ihren eigenen Programmtext
 akzeptiert oder nicht.

Das „universelle“ (allgemeine) Halteproblem

$UNIV = \{ w\$x \in \{0,1\}^*\{0,1\}^* \mid \text{decode}(x) \in L(M_w) \}$

Satz: Die Sprache

$$\text{UNIV} = \{ w\$x \in \{0,1\}^* \$ \{0,1\}^* \mid \text{decod}(x) \in L(M_w) \}$$

ist **nicht** entscheidbar.

Satz: Die Sprache

$$\text{UNIV} = \{ w\$x \in \{0,1\}^* \$ \{0,1\}^* \mid \text{decod}(x) \in L(M_w) \}$$

ist **nicht** entscheidbar.

Beweis:

Annahme, es gäbe einen Entscheider **E** für **UNIV**.

Dies würde einen Entscheider für **SAM** liefern, was unmöglich ist. Also Widerspruch.

Satz: Die Sprache

$$\text{UNIV} = \{ w\$x \in \{0,1\}^* \$ \{0,1\}^* \mid \text{decod}(x) \in L(M_w) \}$$

ist **nicht** entscheidbar.

Beweis:

Annahme, es gäbe einen Entscheider **E** für **UNIV**.

Dies würde einen Entscheider für **SAM** liefern, was unmöglich ist. Also Widerspruch.

$$w \in \text{SAM} \text{ g.d.w. } w\$ \text{cod}(w) \in \text{UNIV}$$

Satz: Die Sprache

$$\text{UNIV} = \{ w\$x \in \{0,1\}^* \$ \{0,1\}^* \mid \text{decod}(x) \in L(M_w) \}$$

ist **nicht** entscheidbar.

Beweis:

Annahme, es gäbe einen Entscheider **E** für **UNIV**.

Dies würde einen Entscheider für **SAM** liefern, was unmöglich ist. Also Widerspruch.

$$w \in \text{SAM} \text{ g.d.w. } w\$ \text{cod}(w) \in \text{UNIV} \\ w\$ \text{cod}(w) \in \text{UNIV} \Leftrightarrow \text{decod}(\text{cod}(w)) \in L(M_w) \Leftrightarrow w \in L(M_w) \Leftrightarrow w \in \text{SAM}$$

Analog: „allgemeines Halteproblem“

Ob eine SML Funktion mit einem vorgegebenen Argument terminiert oder nicht, ist im Allgemeinen **nicht entscheidbar**.

Analog: „allgemeines Halteproblem“

Ob eine SML Funktion mit einem vorgegebenen Argument terminiert oder nicht, ist im Allgemeinen **nicht entscheidbar**.

Das heißt, es gibt kein automatisches Werkzeug, das aus dem Programmtext einer SML Funktion schließen kann, ob diese Funktion ein vorgegebenes Argument akzeptiert oder nicht.

Reduktionen

Def.: Man sagt

“Sprache $A \subseteq \Sigma^*$ reduziert sich auf Sprache $B \subseteq \Gamma^*$ ”

“ A ist leichter als B ”

und schreibt

$A \leq B$,

wenn es eine überall berechenbare Funktion $F: \Sigma^* \rightarrow \Gamma^*$ gibt,
sodass für alle $x \in \Sigma^*$ gilt

$$x \in A \Leftrightarrow F(x) \in B$$

Satz: Wenn $A \leq B$ dann gilt:

(i) B entscheidbar $\Rightarrow A$ entscheidbar

(ii) A nicht entscheidbar $\Rightarrow B$ nicht entscheidbar

Satz: Wenn $A \leq B$ dann gilt:

(i) B entscheidbar $\Rightarrow A$ entscheidbar

(ii) A nicht entscheidbar $\Rightarrow B$ nicht entscheidbar

Beweis: (zeige nur (i), da (i) und (ii) äquivalent)

B entscheidbar, d.h. es gibt Entscheider E für B

$A \leq B$, d.h. es gibt Funktion F , berechnet durch TM M_F
mit $x \in A \Leftrightarrow F(x) \in B$

Entscheide, ob $x \in A$, indem E auf $F(x)$ angewandt wird.

Reduktionen

Def.: Man sagt

“Sprache $A \subseteq \Sigma^*$ reduziert sich auf Sprache $B \subseteq \Gamma^*$ ”

“ A ist leichter als B ”

und schreibt

$A \preceq B$,

wenn es eine überall berechenbare Funktion $F: \Sigma^* \rightarrow \Gamma^*$ gibt, sodass für alle $x \in \Sigma^*$ gilt

$$x \in A \Leftrightarrow F(x) \in B$$

Satz: Wenn $A \preceq B$ dann gilt:

(i) B entscheidbar $\Rightarrow A$ entscheidbar

(ii) A nicht entscheidbar $\Rightarrow B$ nicht entscheidbar

Satz: Wenn $A \preceq B$ dann gilt:

(i) B entscheidbar $\Rightarrow A$ entscheidbar

(ii) A nicht entscheidbar $\Rightarrow B$ nicht entscheidbar

Anwendung: $UNIV$ ist nicht entscheidbar, denn

$SAM \preceq UNIV$, und SAM nicht entscheidbar.

$$SAM = \{ w \in \{0,1\}^* \mid w \in L(M_w) \}$$

$$UNIV = \{ w\$x \in \{0,1\}^*\$ \{0,1\}^* \mid \text{decode}(x) \in L(M_w) \}$$

$$w \xrightarrow{F} w\$ \text{cod}(w) \quad w \in SAM \Leftrightarrow w\$ \text{cod}(w) \in UNIV$$

u beliebiger, fester String

$$ACC_u = \{ w \in \{0,1\}^* \mid u \in L(M_w) \}$$

Satz: Für jedes u ist ACC_u nicht entscheidbar.

u beliebiger, fester String

$$ACC_u = \{ w \in \{0,1\}^* \mid u \in L(M_w) \}$$

Satz: Für jedes u ist ACC_u nicht entscheidbar.

Beweis: $UNIV \preceq ACC_u$ (und $UNIV$ ist nicht entscheidbar)

Reduktionsfunktion: $w\$x$ wird umgewandelt in Beschreibung einer TM $Z_{w,x}$

$Z_{w,x}$: Eingabe y

1. Lasse M_w mit Eingabe $\text{decode}(x)$ laufen.
2. If $y=u$ then akzeptiere, sonst divergiere

u beliebiger, fester String

$$ACC_u = \{ w \in \{0,1\}^* \mid u \in L(M_w) \}$$

Satz: Für jedes u ist ACC_u nicht entscheidbar.

Beweis: $UNIV \preceq ACC_u$ (und $UNIV$ ist nicht entscheidbar)

Reduktionsfunktion: $w\$x$ wird umgewandelt in Beschreibung einer TM $Z_{w,x}$

$Z_{w,x}$: Eingabe y

1. Lasse M_w mit Eingabe $\text{decode}(x)$ laufen.
2. If $y=u$ then akzeptiere, sonst divergiere

$$w\$x \in UNIV \Rightarrow M_w \text{ akzeptiert } \text{decode}(x) \Rightarrow Z_{w,x} \text{ akzeptiert } u \Rightarrow \text{cod}(Z_{w,x}) \in ACC_u$$

$$w\$x \notin UNIV \Rightarrow M_w \text{ akzept. nicht } \text{decode}(x) \Rightarrow Z_{w,x} \text{ akzept. } u \text{ nicht} \Rightarrow \text{cod}(Z_{w,x}) \notin ACC_u$$

Satz von Rice:

$\mathcal{RE}$ Menge aller rekursiv aufzählbaren Sprachen

Für jedes $\mathcal{S} \subset \mathcal{RE}$ mit $\emptyset \neq \mathcal{S} \neq \mathcal{RE}$ gilt

$$C(\mathcal{S}) = \{ w \mid L(M_w) \in \mathcal{S} \}$$

ist nicht entscheidbar.

Satz von Rice:

$\mathcal{RE}$ Menge aller rekursiv aufzählbaren Sprachen

Für jedes $\mathcal{S} \subset \mathcal{RE}$ mit $\emptyset \neq \mathcal{S} \neq \mathcal{RE}$ gilt

$$C(\mathcal{S}) = \{ w \mid L(M_w) \in \mathcal{S} \}$$

ist nicht entscheidbar.

Alle (nicht trivialen) Eigenschaften von Turingmaschinen, die sich durch die **akzeptierte Sprache** ausdrücken lassen, sind **nicht** entscheidbar.

Alle (nicht trivialen) Eigenschaften von Turingmaschinen, die sich durch die **akzeptierte Sprache** ausdrücken lassen, sind **nicht** entscheidbar.

Beispiele:

$$\{ w \mid \varepsilon \in L(M_w) \}$$

$$\{ w \mid L(M) = \emptyset \}$$

$$\{ w \mid L(M) \text{ ist endlich} \}$$

$$\{ w \mid L(M) \text{ ist kontextfrei} \}$$

alle nicht entscheidbar.

Post'sche Korrespondenzproblem

Reservoir von k Spielkartentypen

1	10	011	10
101	00	11	111

Frage: Kann man Spielkarten (mit möglichen Wiederholungen) so nebeneinanderlegen, dass sich **oben** und **unten** das gleiche Wort ergibt?

1	011	10	011
101	11	00	11

Post'sche Korrespondenzproblem PCP_{Σ}

Gegeben: $k \in \mathbb{N}$, $((x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)) \in (\Sigma^*)^{2k}$

Frage: Gibt es $I \in \{1, \dots, k\}^*$ mit $X[I] = Y[I]$

$$I = (i_1, i_2, \dots, i_n)$$

$$X[I] = x_{i_1} x_{i_2} \dots x_{i_n}$$

$$Y[I] = y_{i_1} y_{i_2} \dots y_{i_n}$$

Modifiziertes Post'sche Korrespondenzproblem $MPCP_{\Sigma}$

Gegeben: $k \in \mathbb{N}$, $((x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)) \in (\Sigma^*)^{2k}$

Frage: Gibt es $I \in \{1, \dots, k\}^*$ mit $X[I] = Y[I]$

und $i_1 = 1$

$$I = (i_1, i_2, \dots, i_n)$$

$$X[I] = x_{i_1} x_{i_2} \dots x_{i_n}$$

$$Y[I] = y_{i_1} y_{i_2} \dots y_{i_n}$$

Post'sche Korrespondenzproblem

Reservoir von k Spielkartentypen

1 101	10 00	011 11	10 111
----------	----------	-----------	-----------

Frage: Kann man Spielkarten (mit möglichen Wiederholungen) so nebeneinanderlegen, dass sich oben und unten das gleiche Wort ergibt?

1 101	011 11	10 00	011 11
----------	-----------	----------	-----------

Post'sche Korrespondenzproblem PCP_{Σ}

Gegeben: $k \in \mathbb{N}$, $((x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)) \in (\Sigma^*)^{2k}$

Frage: Gibt es $I \in \{1, \dots, k\}^*$ mit $X[I] = Y[I]$

$$I = (i_1, i_2, \dots, i_n)$$

$$X[I] = x_{i_1} x_{i_2} \dots x_{i_n}$$

$$Y[I] = y_{i_1} y_{i_2} \dots y_{i_n}$$

Modifiziertes Post'sche Korrespondenzproblem $MPCP_{\Sigma}$

Gegeben: $k \in \mathbb{N}$, $((x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)) \in (\Sigma^*)^{2k}$

Frage: Gibt es $I \in \{1, \dots, k\}^*$ mit $X[I] = Y[I]$

und $i_1 = 1$

$$I = (i_1, i_2, \dots, i_n)$$

$$X[I] = x_{i_1} x_{i_2} \dots x_{i_n}$$

$$Y[I] = y_{i_1} y_{i_2} \dots y_{i_n}$$

Beh: $MPCP_{\Sigma} \preceq PCP_{\Sigma \cup \{\#, \$\}}$

Beh: $MPCP_{\Sigma} \preceq PCP_{\Sigma \cup \{\#, \$\}}$

Beweis:

$$(k, ((x_1, y_1), \dots, (x_k, y_k)))$$

Beh: $MPCP_{\Sigma} \preceq PCP_{\Sigma \cup \{\#, \$\}}$

Beweis:

$$(k, ((x_1, y_1), \dots, (x_k, y_k))) \rightarrow (k+2, ((\bar{x}_1, \bar{y}_1), (\acute{x}_1, \acute{y}_1), \dots, (\acute{x}_k, \acute{y}_k), (\$, \$)))$$

Beh: $\text{MPCP}_{\Sigma} \preceq \text{PCP}_{\Sigma \cup \{\#, \$\}}$

Beweis:

$(k, ((x_1, y_1), \dots, (x_k, y_k))) \rightarrow (k+2, ((\bar{x}_1, \bar{y}_1), (\acute{x}_1, \acute{y}_1), \dots, (\acute{x}_k, \acute{y}_k), (\$, \$)))$

$a = a_1 a_2 \dots a_m$

$\bar{a} = \# a_1 \# a_2 \# \dots \# a_m \#$

$\acute{a} = a_1 \# a_2 \# \dots \# a_m \#$

$\`a = \# a_1 \# a_2 \# \dots \# a_m$

WORT: Das Wortproblem

Gegeben: Eine Grammatik $G = (\Sigma, V, S, P)$ und ein Wort $w \in \Sigma^*$

Frage: Ist $w \in L(G)$? Also, wird w von G generiert?

WORT: Das Wortproblem

Gegeben: Eine Grammatik $G = (\Sigma, V, S, P)$ und ein Wort $w \in \Sigma^*$

Frage: Ist $w \in L(G)$? Also, wird w von G generiert?

Als Sprache ausgedrückt:

$\text{WORT} = \{ \text{cod}(G) \$ \text{cod}(w) \mid w \in L(G) \}$

WORT: Das Wortproblem

Gegeben: Eine Grammatik $G = (\Sigma, V, S, P)$ und ein Wort $w \in \Sigma^*$

Frage: Ist $w \in L(G)$? Also, wird w von G generiert?

Als Sprache ausgedrückt:

$\text{WORT} = \{ \text{cod}(G) \$ \text{cod}(w) \mid w \in L(G) \}$

Beh: $\text{UNIV} \preceq \text{WORT}$

WORT: Das Wortproblem

Gegeben: Eine Grammatik $G = (\Sigma, V, S, P)$ und ein Wort $w \in \Sigma^*$

Frage: Ist $w \in L(G)$? Also, wird w von G generiert?

Als Sprache ausgedrückt:

$\text{WORT} = \{ \text{cod}(G) \$ \text{cod}(w) \mid w \in L(G) \}$

Beh: $\text{UNIV} \preceq \text{WORT}$

Beweis schon bei Äquivalenz von TM und Grammatiken gegeben.

Beh: $\text{WORT} \preceq \text{MPCP}$

Beh: $\text{WORT} \preceq \text{MPCP}$

Beweisidee: Gegeben Grammatik $G=(\Sigma,V,S,P)$ und Wort w ,
baue MPCP, dessen Lösung eine Derivation $S \Rightarrow_G w$ darstellt.

$$w \Leftarrow U_m \Leftarrow U_{m-1} \Leftarrow U_{m-2} \Leftarrow \cdots \Leftarrow U_2 \Leftarrow U_1 \Leftarrow S$$

Beh: $\text{WORT} \preceq \text{MPCP}$

Beweisidee: Gegeben Grammatik $G=(\Sigma,V,S,P)$ und Wort w , baue MPCP, dessen Lösung eine Derivation $S \Rightarrow_G w$ darstellt.

$$w \Leftarrow U_m \Leftarrow U_{m-1} \Leftarrow U_{m-2} \Leftarrow \cdots \Leftarrow U_2 \Leftarrow U_1 \Leftarrow S$$

$$W \Leftarrow U_m \Leftarrow U_{m-1} \Leftarrow U_{m-2} \Leftarrow \cdots \Leftarrow U_2 \Leftarrow U_1 \Leftarrow S$$

Beh: $\text{WORT} \preceq \text{MPCP}$

Beweisidee: Gegeben Grammatik $G=(\Sigma,V,S,P)$ und Wort w ,
baue MPCP, dessen Lösung eine Derivation $S \Rightarrow_G w$ darstellt.

$$\mathbb{C}w \Leftarrow U_m \Leftarrow U_{m-1} \Leftarrow U_{m-2} \Leftarrow \cdots \Leftarrow U_2 \Leftarrow U_1 \Leftarrow S \quad \$$$

$$\epsilon \quad w \Leftarrow U_m \Leftarrow U_{m-1} \Leftarrow U_{m-2} \Leftarrow \cdots \Leftarrow U_2 \Leftarrow U_1 \Leftarrow S\$$$

Beh: $\text{WORT} \preceq \text{MPCP}$

Beweisidee: Gegeben Grammatik $G=(\Sigma,V,S,P)$ und Wort w ,
baue MPCP, dessen Lösung eine Derivation $S \Rightarrow_G w$ darstellt.

$$\begin{array}{cccccccccccccccc} \text{\textcolor{red}{€}}w & \Leftarrow & U_m & \Leftarrow & U_{m-1} & \Leftarrow & U_{m-2} & \Leftarrow & \dots & \Leftarrow & U_2 & \Leftarrow & U_1 & \Leftarrow & S & \text{\textcolor{red}{\$}} \\ \boxed{\text{\textcolor{red}{€}}} & & w & \Leftarrow & U_m & \Leftarrow & U_{m-1} & \Leftarrow & U_{m-2} & \Leftarrow & \dots & \Leftarrow & U_2 & \Leftarrow & U_1 & \Leftarrow & S\$ \end{array}$$

0	0	1	A	0	0	1	B	1	1
0	0	1	A	0	0	1	A	0	1

Bei Produktion
 $B \rightarrow A0$
 in P

Beh: $\text{WORT} \preceq \text{MPCP}$

Beweisidee: Gegeben Grammatik $G=(\Sigma,V,S,P)$ und Wort w ,
baue MPCP, dessen Lösung eine Derivation $S \Rightarrow_G w$ darstellt.

$$\begin{array}{cccccccccccccccccccc} \boxed{\text{€w}} & \Leftarrow & \boxed{\text{U}_m} & \Leftarrow & \boxed{\text{U}_{m-1}} & \Leftarrow & \boxed{\text{U}_{m-2}} & \Leftarrow & \cdots & \Leftarrow & \boxed{\text{U}_2} & \Leftarrow & \boxed{\text{U}_1} & \Leftarrow & \boxed{\text{S}} & \Leftarrow & \boxed{\text{\$}} \\ \boxed{\text{€}} & & \boxed{\text{w}} & \Leftarrow & \boxed{\text{U}_m} & \Leftarrow & \boxed{\text{U}_{m-1}} & \Leftarrow & \boxed{\text{U}_{m-2}} & \Leftarrow & \cdots & \Leftarrow & \boxed{\text{U}_2} & \Leftarrow & \boxed{\text{U}_1} & \Leftarrow & \boxed{\text{S\$}} \end{array}$$

Grammatik $G = (\{a_1, \dots, a_s\}, \{B_1, \dots, B_t\}, B_1, \{ \alpha_1 \rightarrow \beta_1, \dots, \alpha_r \rightarrow \beta_r \}),$ Wort w

Beh: $\text{WORT} \preceq \text{MPCP}$

Beweisidee: Gegeben Grammatik $G=(\Sigma,V,S,P)$ und Wort w ,
baue MPCP, dessen Lösung eine Derivation $S \Rightarrow_G w$ darstellt.

$$\begin{array}{cccccccccccccccc} \text{\textcolor{red}{€}}w & \Leftarrow & U_m & \Leftarrow & U_{m-1} & \Leftarrow & U_{m-2} & \Leftarrow & \dots & \Leftarrow & U_2 & \Leftarrow & U_1 & \Leftarrow & S & \text{\textcolor{red}{\$}} \\ \boxed{\text{\textcolor{red}{€}}} & & w & \Leftarrow & U_m & \Leftarrow & U_{m-1} & \Leftarrow & U_{m-2} & \Leftarrow & \dots & \Leftarrow & U_2 & \Leftarrow & U_1 & \Leftarrow & S\$ \end{array}$$

Grammatik $G = (\{a_1, \dots, a_n\}, \{B_1, \dots, B_m\}, B_1, \{ \alpha_1 \rightarrow \beta_1, \dots, \alpha_r \rightarrow \beta_r \})$, Wort w

MPCP

$$\begin{array}{|c|} \hline \textcolor{red}{\epsilon} w \Leftarrow \\ \hline \epsilon \\ \hline \end{array} \quad \begin{array}{|c|} \hline \Leftarrow \\ \hline \Leftarrow \\ \hline \end{array} \quad \begin{array}{|c|} \hline a_i \\ \hline a_i \\ \hline \end{array} \quad \dots \quad \begin{array}{|c|} \hline a_s \\ \hline a_s \\ \hline \end{array} \quad \begin{array}{|c|} \hline B_1 \\ \hline B_1 \\ \hline \end{array} \quad \dots \quad \begin{array}{|c|} \hline B_t \\ \hline B_t \\ \hline \end{array} \quad \begin{array}{|c|} \hline \textcolor{red}{\alpha}_1 \\ \hline \beta_1 \\ \hline \end{array} \quad \dots \quad \begin{array}{|c|} \hline \textcolor{red}{\alpha}_r \\ \hline \beta_r \\ \hline \end{array} \quad \begin{array}{|c|} \hline \textcolor{red}{\$} \\ \hline \Leftarrow \$\$ \\ \hline \end{array}$$

Satz: PCP ist nicht entscheidbar.

Satz: PCP ist nicht entscheidbar.

Beweis:

$$\text{UNIV} \preceq \text{WORT} \preceq \text{MPCP} \preceq \text{PCP},$$

also

$$\text{UNIV} \preceq \text{PCP}.$$

Da UNIV unentscheidbar, ist also auch PCP unentscheidbar.

Satz: Die folgenden Probleme sind alle unentscheidbar:

Gegeben: kontextfreie Grammatiken G_1 und G_2

- 1) Frage: Gilt $L(G_1) \cap L(G_2) = \emptyset$?
- 2) Frage: Gilt $L(G_1) \cap L(G_2)$ ist endlich ?
- 3) Frage: Gilt $L(G_1) \subset L(G_2)$?
- 4) Frage: Gilt $L(G_1) = L(G_2)$?
- 5) Frage: Gilt $L(G_1) \cap L(G_2)$ ist kontextfrei ?

Satz: Die folgenden Probleme sind alle unentscheidbar:

Gegeben: kontextfreie Grammatiken G_1 und G_2

- 1) Frage: Gilt $L(G_1) \cap L(G_2) = \emptyset$?
- 2) Frage: Gilt $L(G_1) \cap L(G_2)$ ist endlich ?
- 3) Frage: Gilt $L(G_1) \subset L(G_2)$?
- 4) Frage: Gilt $L(G_1) = L(G_2)$?
- 5) Frage: Gilt $L(G_1) \cap L(G_2)$ ist kontextfrei ?

Anmerkung: Die analogen Fragen für reguläre Sprachen sind alle entscheidbar.

Beh.: Gilt $L(G_1) \cap L(G_2) = \emptyset$? ist unentscheidbar.

Beh.: Gilt $L(G_1) \cap L(G_2) = \emptyset$? ist unentscheidbar.

Beweis: $(k, ((x_1, y_1), \dots, (x_k, y_k)))$ Instanz von PCP

Betrachte Sprachen $S_X = \{ I^R X[I] \mid I \in \{1, \dots, k\}^+ \}$
 $S_Y = \{ J^R Y[J] \mid J \in \{1, \dots, k\}^+ \}$

$$X[(i_1, \dots, i_m)] = x_{i_1} \dots x_{i_m}$$

Beh.: Gilt $L(G_1) \cap L(G_2) = \emptyset$? ist unentscheidbar.

Beweis: $(k, ((x_1, y_1), \dots, (x_k, y_k)))$ Instanz von PCP

Betrachte Sprachen $S_X = \{ I^R \$ X[I] \mid I \in \{1, \dots, k\}^+ \}$
 $S_Y = \{ J^R \$ Y[J] \mid J \in \{1, \dots, k\}^+ \}$

$$X[(i_1, \dots, i_m)] = x_{i_1} \dots x_{i_m}$$

S_X kontextfrei, denn generiert durch Grammatik mit Regeln

$$\begin{aligned} S &\rightarrow i \ S \ x_i & 1 \leq i \leq k \\ S &\rightarrow i \ \$ \ x_i & 1 \leq i \leq k \end{aligned}$$

analog S_Y kontextfrei.

S_X und S_Y sind sogar **deterministisch** kontextfrei.

Beh.: Gilt $L(G_1) \cap L(G_2) = \emptyset$? ist unentscheidbar.

Beweis: $(k, ((x_1, y_1), \dots, (x_k, y_k)))$ Instanz von PCP

Betrachte Sprachen $S_X = \{ I^R \$ X[I] \mid I \in \{1, \dots, k\}^+ \}$
 $S_Y = \{ J^R \$ Y[J] \mid J \in \{1, \dots, k\}^+ \}$

$$X[(i_1, \dots, i_m)] = x_{i_1} \dots x_{i_m}$$

$S_X \cap S_Y \neq \emptyset$ bedeutet es gibt ein $I \in \{1, \dots, k\}^+$ mit

$$I^R \$ X[I] = I^R \$ Y[I], \text{ also } X[I] = Y[I],$$

also entscheiden ob $S_X \cap S_Y \neq \emptyset$ würde das PCP lösen.

Post'sche Korrespondenzproblem

Reservoir von k Spielkartentypen

1	10	011	10
101	00	11	111

Frage: Kann man Spielkarten (mit möglichen Wiederholungen) so nebeneinanderlegen, dass sich oben und unten das gleiche Wort ergibt ?

1	011	10	011
101	11	00	11

Post'sche Korrespondenzproblem PCP_{Σ}

Gegeben: $k \in \mathbb{N}$, $((x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)) \in (\Sigma^*)^{2k}$

Frage: Gibt es $I \in \{1, \dots, k\}^*$ mit $X[I] = Y[I]$

$I = (i_1, i_2, \dots, i_n)$

$X[I] = x_{i_1} x_{i_2} \dots x_{i_n}$

$Y[I] = y_{i_1} y_{i_2} \dots y_{i_n}$

Satz: PCP ist nicht entscheidbar.

Beweis:

$UNIV \preceq WORT \preceq MPCP \preceq PCP$,

also

$UNIV \preceq PCP$.

Da UNIV unentscheidbar, ist also auch PCP unentscheidbar.

Satz: Die folgenden Probleme sind alle unentscheidbar:

Gegeben: kontextfreie Grammatiken G_1 und G_2

- 1) Frage: Gilt $L(G_1) \cap L(G_2) = \emptyset$?
- 2) Frage: Gilt $L(G_1) \cap L(G_2)$ ist endlich ?
- 3) Frage: Gilt $L(G_1) \subset L(G_2)$?
- 4) Frage: Gilt $L(G_1) = L(G_2)$?
- 5) Frage: Gilt $L(G_1) \cap L(G_2)$ ist kontextfrei ?

Satz: Die folgenden Probleme sind alle unentscheidbar:

Gegeben: kontextfreie Grammatiken G_1 und G_2

- 1) Frage: Gilt $L(G_1) \cap L(G_2) = \emptyset$?
- 2) Frage: Gilt $L(G_1) \cap L(G_2)$ ist endlich ?
- 3) Frage: Gilt $L(G_1) \subset L(G_2)$?
- 4) Frage: Gilt $L(G_1) = L(G_2)$?
- 5) Frage: Gilt $L(G_1) \cap L(G_2)$ ist kontextfrei ?

Anmerkung: Die analogen Fragen für reguläre Sprachen sind alle entscheidbar.

Beh.: Gilt $L(G_1) \cap L(G_2) = \emptyset$? ist unentscheidbar.

Beh.: Gilt $L(G_1) \cap L(G_2) = \emptyset$? ist unentscheidbar.

Beweis: $(k, ((x_1, y_1), \dots, (x_k, y_k)))$ Instanz von PCP

Betrachte Sprachen $S_X = \{ I^R \$ X[I] \mid I \in \{1, \dots, k\}^+ \}$
 $S_Y = \{ J^R \$ Y[J] \mid J \in \{1, \dots, k\}^+ \}$

$$X(i_1, \dots, i_m) = x_{i_1} \dots x_{i_m}$$

Beh.: Gilt $L(G_1) \cap L(G_2) = \emptyset$? ist unentscheidbar.

Beweis: $(k, ((x_1, y_1), \dots, (x_k, y_k)))$ Instanz von PCP

Betrachte Sprachen $S_X = \{ I^R \$ X[I] \mid I \in \{1, \dots, k\}^+ \}$
 $S_Y = \{ J^R \$ Y[J] \mid J \in \{1, \dots, k\}^+ \}$

$$X(i_1, \dots, i_m) = x_{i_1} \dots x_{i_m}$$

S_X kontextfrei, denn generiert durch Grammatik mit Regeln

$$\begin{aligned} S &\rightarrow i S x_i & 1 \leq i \leq k \\ S &\rightarrow i \$ x_i & 1 \leq i \leq k \end{aligned}$$

analog S_Y kontextfrei.

S_X und S_Y sind sogar **deterministisch** kontextfrei.

Beh.: Gilt $L(G_1) \cap L(G_2) = \emptyset$? ist unentscheidbar.

Beweis: $(k, ((x_1, y_1), \dots, (x_k, y_k)))$ Instanz von PCP

Betrachte Sprachen $S_X = \{ I^R \$ X[I] \mid I \in \{1, \dots, k\}^+ \}$
 $S_Y = \{ J^R \$ Y[J] \mid J \in \{1, \dots, k\}^+ \}$

$$X(i_1, \dots, i_m) = x_{i_1} \dots x_{i_m}$$

$S_X \cap S_Y \neq \emptyset$ bedeutet es gibt ein $I \in \{1, \dots, k\}^+$ mit

$$I^R \$ X[I] = I^R \$ Y[I], \text{ also } X[I] = Y[I],$$

also entscheiden, ob $S_X \cap S_Y \neq \emptyset$ würde das PCP lösen.

Beh.: Gilt $L(G_1) \cap L(G_2)$ ist endlich? ist unentscheidbar.

Beweis: $(k, ((x_1, y_1), \dots, (x_k, y_k)))$ Instanz von PCP

Betrachte Sprachen $S_X = \{ I^R \$ X[I] \mid I \in \{1, \dots, k\}^+ \}$
 $S_Y = \{ J^R \$ Y[J] \mid J \in \{1, \dots, k\}^+ \}$

PCP hat keine Lösung $\Rightarrow S_X \cap S_Y = \emptyset$, also $S_X \cap S_Y$ endlich

PCP hat Lösung $I \Rightarrow I^n$ ist Lösung für jedes $n > 0$

$$\begin{aligned} &\Rightarrow (I^n)^R \$ X[I^n] = (I^n)^R \$ Y[I^n] \text{ jedes } n > 0 \\ &\Rightarrow S_X \cap S_Y \text{ ist nicht endlich.} \end{aligned}$$

Satz: Die folgenden Probleme sind alle unentscheidbar:

Gegeben: kontextfreie Grammatiken G_1 und G_2

- 1) Frage: Gilt $L(G_1) \cap L(G_2) = \emptyset$?
- 2) Frage: Gilt $L(G_1) \cap L(G_2)$ ist endlich ?
- 3) Frage: Gilt $L(G_1) \subset L(G_2)$?
- 4) Frage: Gilt $L(G_1) = L(G_2)$?
- 5) Frage: Gilt $L(G_1) \cap L(G_2)$ ist kontextfrei ?

Anmerkung: Die analogen Fragen für reguläre Sprachen sind alle entscheidbar.

Beh.: Gilt $L(G_1) \subset L(G_2)$? ist unentscheidbar.

Beweis: $(k, ((x_1, y_1), \dots, (x_k, y_k)))$ Instanz von PCP

Betrachte Sprachen $S_X = \{ I^R \$ X[I] \mid I \in \{1, \dots, k\}^+ \}$
 $S_Y = \{ J^R \$ Y[J] \mid J \in \{1, \dots, k\}^+ \}$

G_1 kontextfreie Grammatik für S_X

G_2 kontextfreie Grammatik für S_Y

(geht, da S_Y deterministisch kontextfrei)

$$L(G_1) \subset L(G_2) \Leftrightarrow S_X \subset \overline{S_Y} \Leftrightarrow S_X \cap S_Y = \emptyset \Leftrightarrow \underbrace{\text{PCP unlösbar}}_{\text{unentscheidbar}}$$

Satz: Die folgenden Probleme sind alle unentscheidbar:

Gegeben: kontextfreie Grammatiken G_1 und G_2

- 1) Frage: Gilt $L(G_1) \cap L(G_2) = \emptyset$?
- 2) Frage: Gilt $L(G_1) \cap L(G_2)$ ist endlich ?
- 3) Frage: Gilt $L(G_1) \subset L(G_2)$?
- 4) Frage: Gilt $L(G_1) = L(G_2)$?
- 5) Frage: Gilt $L(G_1) \cap L(G_2)$ ist kontextfrei ?

Anmerkung: Die analogen Fragen für reguläre Sprachen sind alle entscheidbar.

Beh.: Gilt $L(G_1) = L(G_2)$? ist unentscheidbar.

Beweis: $(k, ((x_1, y_1), \dots, (x_k, y_k)))$ Instanz von PCP

Betrachte Sprachen $S_X = \{ I^R \$ X [I] \mid I \in \{1, \dots, k\}^+ \}$
 $S_Y = \{ J^R \$ Y [J] \mid J \in \{1, \dots, k\}^+ \}$

G_1 kontextfreie Grammatik für S_X

G_2 kontextfreie Grammatik für S_Y

(geht, da S_Y deterministisch kontextfrei)

G kontextfreie Grammatik für $S_X \cup S_Y$

$L(G_2) = L(G) \Leftrightarrow L(G_1) \subset L(G_2) \Leftrightarrow S_X \subset \overline{S_Y} \Leftrightarrow S_X \cap S_Y = \emptyset \Leftrightarrow \text{PCP unlösbar}$
unentscheidbar

Satz: Die folgenden Probleme sind alle unentscheidbar:

Gegeben: kontextfreie Grammatiken G_1 und G_2

- 1) Frage: Gilt $L(G_1) \cap L(G_2) = \emptyset$?
- 2) Frage: Gilt $L(G_1) \cap L(G_2)$ ist endlich ?
- 3) Frage: Gilt $L(G_1) \subset L(G_2)$?
- 4) Frage: Gilt $L(G_1) = L(G_2)$?
- 5) Frage: Gilt $L(G_1) \cap L(G_2)$ ist kontextfrei ?

Anmerkung: Die analogen Fragen für reguläre Sprachen sind alle entscheidbar.

Beh.: Gilt $L(G_1) \cap L(G_2)$ ist kontextfrei? ist unentscheidbar.

Beweis: $(k, ((x_1, y_1), \dots, (x_k, y_k)))$ Instanz von PCP

Betrachte Sprachen $L_X = \{ I^R \# K \$ K^R \# X [I] \mid I, K \in \{1, \dots, k\}^+ \}$
 $S_Y = \{ J^R \# J \$ H^R \# Y [H] \mid J, H \in \{1, \dots, k\}^+ \}$

G_1 kontextfreie Grammatik für L_X

G_2 kontextfreie Grammatik für L_Y

PCP unlösbar $\Rightarrow L_X \cap L_Y = \emptyset \Rightarrow L_X \cap L_Y$ ist kontextfrei

PCP lösbar $\Rightarrow L_X \cap L_Y$ enthält Worte der Form $(I^R)^R \# I^R \$ (I^R)^R \# X [I^R]$
 für alle $n > 0$
 $\Rightarrow L_X \cap L_Y$ ist nicht kontextfrei