

1.9 Das Postsche Korrespondenz-Problem

Definition.

Das nachfolgend beschriebene Problem heißt *Postsches Korrespondenz-Problem* (PCP).

gegeben: Eine endliche Folgen $(x_1, y_1), \dots, (x_k, y_k)$ von Wortpaaren mit $x_i, y_i \in A^+$ (A endliche Alphabet)
 gefragt: Gibt es eine Folge von Indizes $i_1, \dots, i_n \in \{1, \dots, k\}$, $n \geq 1$, mit $x_{i_1}, \dots, x_{i_n} = y_{i_1}, \dots, y_{i_n}$?

Beispiel.

Das Postsche Korrespondenz-Problem

$$K = ((1, 101), (10, 00), (011, 11)),$$

also

$$\begin{array}{lll} x_1 = 1 & x_2 = 10 & x_3 = 011 \\ y_1 = 101 & y_2 = 00 & y_3 = 11 \end{array}$$

besitzt die Lösung $(1, 3, 2, 3)$, denn es gilt:

$$x_1 x_3 x_2 x_3 = 101110011 = y_1 y_3 y_2 y_3$$

PCP ist schwer.

z.B.: $K = ((001, 0), (01, 011), (01, 101), (10, 001))$

besitzt eine Lösung, aber die kürzeste besteht aus 66 Indizes!

PCP ist semi-entscheidbar:

Probiere alle möglichen Indexfolgen mit zunehmender Länge durch.

Satz.

Das Postsche Korrespondenz-Problem ist nicht entscheidbar.

Beweis.

Betrachte modifiziertes PCP:

MPCP

gegeben: wie bei PCP
 gefragt: gibt es eine Lösung $i_1, \dots, i_n$ mit $i_1 = 1$

Lemma

$\text{MPCP} \leq \text{PCP}$

Beweis.

Sei $\#$ ein neues Symbol.

Für $w = a_1 \dots a_m \in A^+$ sei

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

$$\begin{aligned}\dot{w} &= \#a_1\#a_2\#\dots\#a_m \\ \acute{w} &= a_1\#a_2\#\dots\#a_m\#\end{aligned}$$

Sei $K = ((x_1, y_1), \dots, (x_k, y_k))$ die Eingabe von MPCP.
 $\rightarrow f(K) = ((\bar{x}_1, \dot{y}_1), (x'_1, \dot{y}_1), (x'_2, \dot{y}_2), \dots, (x'_k, \dot{y}_k), (\$, \#\$))$
 f ist berechenbar.
 f vermittelt eine Reduktion von MPCP auf PCP.

Behauptung.

K besitzt eine Lösung mit $i_1 = 1$
 $\Leftrightarrow f(K)$ besitzt irgend eine Lösung.

Beweis.

- ($\rightarrow$) Besitzt K eine Lösung $(i_1, \dots, i_n)$ mit $i_1 = 1$,
dann ist $(1, i_2 + 1, \dots, i_n + 1, k + 2)$ eine Lösung für $f(K)$.
($\leftarrow$) Besitzt $f(K)$ eine Lösung $(i_1, \dots, i_n)$,
so muß gelten:
 $i_1 = 1, i_n = k + 2$ mit $i_j \in \{2, \dots, k + 1\}$ für $2 \leq j \leq n + 1$
 $\Rightarrow (1, i_2 - 1, \dots, i_{n-1} - 1)$ ist Lösung für K .

Zur Unentscheidbarkeit von MPCP:

Lemma

$H \leq MPCP$

Beweis.

Sei $M = (Q, \Sigma, \Gamma, \delta, z_0, \square, F)$ eine kodierte TM

$w \in \Sigma^+$ die Eingabe.

Suche eine allgemeine Vorschrift, die (M, w) eine Folge $(x_1, y_1), \dots, (x_n, y_n)$ zuordnet mit

M angesetzt auf w stoppt $\Leftrightarrow (x_1, y_1), \dots, (x_n, y_n)$ besitzt eine Lösung mit $i_1 = 1$

Konstruktion von MPCP über dem Alphabet $\Gamma \cup Q \cup \{\#\}$.

erstes Wortpaar: $(\#, \#q_0w\#)$

weitere Paare:

1. Kopieregeln:

$$(a, a) \text{ für alle } a \in \Gamma \cup \{\#\}$$

2. Überführungsregeln:

$$\begin{aligned}(qa, q'c) & \text{ falls } \delta(q, a) = (q', c, N) \\ (qa, cq') & \text{ falls } \delta(q, a) = (q', c, R)\end{aligned}$$

$$\begin{aligned}
(qa, q'bc) & \text{ falls } \delta(q, a) = (q', c, L) \text{ für alle } b \in \Gamma \\
(\#qa, \#q'\square c) & \text{ falls } \delta(q, a) = (q', c, L) \\
(q\#, q'c\#) & \text{ falls } \delta(q, \square) = (q', c, N) \\
(q\#, cq'\#) & \text{ falls } \delta(q, \square) = (q', c, R) \\
(bq\#, q'b\#) & \text{ falls } \delta(q, \square) = (q', c, L) \text{ für alle } b \in \Gamma
\end{aligned}$$

3. Löschregeln:

$$(aq_e, q_e) \text{ und } (q_e a, q_e) \text{ für alle } a \in \Gamma, q_e \in F$$

4. Abschlußregeln:

$$(q_e \# \# \#, \#) \text{ für alle } q_e \in F$$

- ($\rightarrow$) Falls M bei w stoppt gibt es eine Folge von Konfigurationen $(k_0, k_1, \dots, k_t)$ mit $k_0 = q_0 w$, k_t ist Endkonfiguration also $k_t = u q_e v$ mit $u, v \in A^*$, $q_e \in F$) und $k_i \vdash k_{i+1}$
 $\Leftrightarrow$ Die obige Eingabe für das MPCP besitzt ein Lösungswort

$$\#k_0\#k_1\#\dots\#k_t\#k'_t\#k''_t\#\dots\#q_e\#\#$$

($k'_t, k''_t, \dots$ entstehen aus $k_t = u q_e v$ durch Löschung von Nachbarsymbolen aus q_e)

- ($\leftarrow$) Besitzt der obige MPCP eine Lösung mit $i_1 = 1$, dann läßt sich eine stoppende Rechnung von M bei w ablesen. ■

Folgerung

PCP bleibt unentscheidbar, falls für das Alphabet A gilt: $A = \{0, 1\}$. ("01-PCP")

Beweis.

Es reicht zu zeigen: $\text{PCP} \leq 01\text{-PCP}$.

Sei A das Alphabet von PCP.

Betrachte: $f : A \rightarrow \{0, 1\}^*$ mit $f(a_r) = \hat{a}_r = 01^r$.

Setze die Abbildung f auf A^* fort.

Es gilt:

$$(x_1, y_1), \dots, (x_n, y_n) \text{ besitzt Lösung} \Leftrightarrow (\hat{x}_1, \hat{y}_1), \dots, (\hat{x}_n, \hat{y}_n) \text{ besitzt Lösung.} \quad \blacksquare$$

Bemerkung

Sei PCP_k die Variante des PCPs, deren Eingabe aus genau k Wortpaaren besteht.

Es gilt:

PCP_k ist unentscheidbar für $k \geq 9$.

PCP_k ist entscheidbar für $k \leq 2$.

offen sonst.

Folgerung

Das Halte-Problem H für TM ist semi-entscheidbar.

Beweis.

PCP ist semi-entscheidbar,

$H \leq \text{PCP}$.

$\Rightarrow$ Beh. ■

Universelle TM: Nachobiger Folgerung gibt es eine TM U , die sich bei Eingabe von $w\#x$ so verhält wie M_w bei Eingabe von x . (Zunächst nur im Bezug auf Halten, bzw. Nicht-Halten)

TM U verhält sich wie ein TM-Interpreter

Der erste Teil programmiert die TM.

Der zweite Teil ist die eigentliche Eingabe.

Man kann mit Hilfe des PCP-Resultats die Unentscheidbarkeit von Problemen der Theorie der formalen Sprachen nachweisen:

Satz.

Das Schnittproblem für kontextfreie Sprachen (G_1, G_2 kontextfreie Sprachen. Gilt: $L(G_1) \cap L(G_2) = \emptyset$?) ist unentscheidbar.

Beweis.

Zu zeigen: $\text{PCP} \leq \text{Schnittproblem}$

Jedem PCP $K = ((x_1, y_1), \dots, (x_k, y_k))$ müssen effektiv zwei Grammatiken G_1, G_2 zugeordnet werden,

so daß gilt:

K besitzt eine Lösung

$\Leftrightarrow$ es gibt ein $w \in L(G_1) \cap L(G_2)$

Idee: Erzeugung von Folgen von x_i -Wörtern mit G_1 .

Erzeugung von Folgen von y_i -Wörtern mit G_2 .

Die Indizes der x_i müssen mit denen der y_i in allen potentiellen Lösungen übereinstimmen.

Sei $G_i = (\{S_i\}, A \cup \{a_1, \dots, a_k\}, P_i, S_i)$ $i = 1, 2$

mit

$$P_1 = \{S_1 \rightarrow a_1 x_1 | \dots | a_k x_k\} \cup \{S_1 \rightarrow a_1 S_1 x_1 | \dots | a_k S_1 x_k\}$$

$$P_2 = \{S_2 \rightarrow a_1 y_1 | \dots | a_k y_k\} \cup \{S_2 \rightarrow a_1 S_2 y_1 | \dots | a_k S_2 y_k\}$$

Es gilt:

K besitzt die Lösung $i_1, \dots, i_n$

$$\Leftrightarrow$$

$$a_{i_n} \dots a_{i_2} a_{i_1} x_{i_1} x_{i_2} \dots x_{i_n} =$$

$$a_{i_n} \dots a_{i_2} a_{i_1} y_{i_1} y_{i_2} \dots y_{i_n} \in L(G_1) \cap L(G_2) \quad \blacksquare$$

Folgerung

Das Schnittproblem für deterministisch kontextfreie Sprachen ist unentscheidbar.

Beweis.

G_1, G_2 aus dem Beweis des letzten Satz es sind deterministisch. $\blacksquare$

Satz.

Das Äquivalenzproblem für kontextfreie Sprachen ist unentscheidbar.

Beweis.

Reduzierung des Schnittproblems für kontextfreie Sprachen auf das Äquivalenzproblem für Kontextfreie Sprachen.

Nutzung von

- deterministisch kontextfreie Sprachen sind effektiv unter Komplementbildung abgeschlossen.
- deterministisch kontextfreie Sprachen sind effektiv unter Vereinigung abgeschlossen.

Es gilt: (G_1, G_2) Schnittproblem

$$\Leftrightarrow L(G_1) \cap L(G_2) = \emptyset$$

$$\Leftrightarrow L(G_1) \subseteq \overline{L(G_2)}$$

$$\Leftrightarrow L(G_1) \subseteq L(G'_2) \quad G'_2 \text{ Grammatik des Komplements von } G_2$$

$$\Leftrightarrow L(G_1) \cup L(G_2) = L(G'_2)$$

$$\Leftrightarrow L(G_3) = L(G_2) \quad G_3 \text{ Vereinigungsgrammatik von } G_1 \text{ und } G_2$$

$$\Leftrightarrow (G_3, G'_2) \in \text{Äquivalenzproblem} \quad \blacksquare$$

noch offen: Das Äquivalenzproblem für deterministisch kontextfreie Grammatiken.

Folgerung

Das Äquivalenzproblem für folgende Probleme ist unentscheidbar:

- nichtdeterministische Kellerautomaten
- BNF
- EBNF
- Syntaxdiagramme

- LBA
- kontextsensitive Grammatiken
- TM

Beweis.

Man kann kontextfreie Grammatiken effektiv in die genannten Probleme übersetzen. ■

Satz.

Das Leerheitsproblem für kontextsensitive Sprachen ist unentscheidbar.

Beweis.

Das Schnittproblem für kontextfreie Sprachen ist entscheidbar.

$\Leftrightarrow (G_1, G_2) \mapsto G_3$ ist berechenbar, wobei

$$L(G_3) = L(G_1) \cap L(G_2)$$

Diese Abbildung vermittelt die Reduktion des Schnittproblems für kontextfreie Sprachen auf das Leerheitsproblem für kontextsensitive Sprachen. ■