

$$MPCP \leq PCP$$

$$H \leq MPCP$$

Lemma $H \leq MPCP$

Bew: Gegeben eine Kodierung einer TM $M = (Z, \Sigma, \Gamma, \delta, z_0, \sqcup, F)$
und Eingabewort $w \in \Sigma^*$, konstruiere MPCP K , so dass

$K = ((x_1, y_1), \dots, (x_n, y_n))$ und eine Lösung

$(i_1, \dots, i_n)$ mit $i_1 = 1$

folgt.

M angeseht auf w hält

MPCP über Alphabet $\Gamma \cup Z \cup \{\#\}$, wobei $\# \notin \Gamma$

Konstruiere folgende Paare

Startpaar:

$$\underline{(\#, \# z_0 w \#)} \quad \text{mit Index 1}$$

Kopierpaare:

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

Überführungspaare:

$$(z a, z' c) \quad \text{falls } \delta(z, a) = (z', c, N) \quad \text{für alle } z, z' \in Z, a, c \in \Gamma$$

$$\underline{(z a, c z')} \quad \text{falls } \delta(z, a) = (z', c, R)$$

$$(b z a, z' b c) \quad \text{falls } \delta(z, a) = (z', c, L)$$

$$(\# z a, \# z' \sqcap c) \quad \text{falls } \delta(z, a) = (z', c, L)$$

$$(z \#, z c \#) \quad \text{falls } \delta(z, \sqcap) = (z', c, N)$$

$$(z \#, c z' \#) \quad \text{falls } \delta(z, \sqcap) = (z', c, R)$$

$$(b z \#, z' b c \#) \quad \text{falls } \delta(z, \sqcap) = (z', c, L)$$

Bsp

$$\delta(z_0, a_1) = (b, z_1, R)$$

$$x = \# z_0 a_1 a_2 \dots a_n \#$$

$$y = \# z_0 a_1 \dots a_n \# \quad \underline{b z_1 a_2 \dots a_n \#}$$

Löschpaare:

$$\begin{array}{ccc} (a z_e, z_e) & \text{für alle } a \in T \text{ und } z_e \in E \\ (z_e^a, z_e) & \text{--- " ---} \end{array}$$

Abschlusspaar

$$(z_e \#\#, \#) \quad \text{für alle } z_e \in E.$$

OBdA aus Endzuständen gibt es keine abgehende Kante.

Stoppt M auf w , so gibt es eine Konfigurationsfolge

$(k_0, k_1, \dots, k_t)$ und $k_0 = z_0 w$ und k_t ist Endkonf.

mit $k_i = v z_e v$, $v, v \in T^*$, $z_e \in E$ und $k_i \vdash k_{i+1}$.

Dann ex. ein Lösungsword für das MPCP:

$$\# k_0 \# k_1 \# \dots \# k_t \# k_t' \# k_t'' \# \dots \# z_e \# \#$$

Umgekehrt: Hat man MPCP eine Lösung, dann kann

man daraus eine Rechnung der TM M ableiten.

D.h. M stoppt auf w . D.h. es gilt $H \in \text{MPCP}$. $\square$

Satz PCP ist unentscheidbar.

Bew: Da $H \in \text{MPCP}$ und $\text{MPCP} \subseteq \text{PCP}$ und da $\subseteq$ transitiv ist, folgt $H \in \text{PCP}$. Da H unentscheidbar ist, muss auch PCP unentscheidbar sein. 

Bem: Üben Sie Reduktionen!

Bem: $\text{PCP} \subseteq H$ bringt nichts!

Bem: Da PCP semi-entscheidbar, muss auch H semi-entscheidbar sein.

TMs, die andere TM simulieren, nennt man universelle Turing-Maschinen.

$$H = \{x\#y \mid M_x \text{ hält auf } y\}$$

Die TM, die H semi-entscheidet, ist die universelle TM.

Satz

Das PCP ist bereits unentscheidbar, wenn das Alphabet $\Sigma = \{0, 1\}$ ist.

Bew:

PCP über $\{0, 1\}$ nennen wir 01-PCP.

$$\begin{aligned} & \cancel{01\text{-PCP} \in \text{PCP}} \\ & \boxed{\text{PCP} \in 01\text{-PCP}} \end{aligned}$$

Sei $\Sigma = \{a_1, \dots, a_n\}$ das Alphabet unseres PCPs. Jedem $a_j \in \Sigma$ ordne wir ein $\hat{a}_j \in \{0, 1\}^k$ zu. $\hat{a}_j = 01^j$. Für ein Wort $w = a_{i_1} \dots a_{i_n}$ sei

$$\hat{w} = \hat{a}_{i_1} \dots \hat{a}_{i_n}.$$

Offensichtlich

$(i_1, \dots, i_n)$ ist eine Lösung des PCPs gdw.

$(i_1, \dots, i_n)$ ist eine Lösung des 01-PCPs.

□

PCPs über $\Sigma = \{1\}$

Aussage PCPs über $\Sigma = \{1\}$ sind entscheidbar.

Bew: Gegeben $K = ((x_1, y_1), \dots, (x_k, y_k))$ $x_i, y_i \in \{1\}^*$

Falls $|x_i| < |y_i|$ für alle $i \in \{1, \dots, k\}$, dann ist es
keine Lösung

Falls $|x_i| > |y_i|$ — " — — " —

Falls $|x_i| = |y_i|$ für mind. ein $i \in \{1, \dots, k\}$, dann ist die
Folge (i) eine

Falls $|x_i| > |y_i|$ und $|x_j| < |y_j|$ für $i, j \in \{1, \dots, k\}$, dann
ist es folgende Lösung (Lösung)
 $\{d_i := |x_i| - |y_i|, d_j := |y_j| - |x_j|\}$

$$\underbrace{(i, \dots, i)}_{\lg(|d_i|, d_j)/d_j}, \underbrace{(j, \dots, j)}_{\lg(|d_i|, d_j)/d_i}$$

Bem. Sei PCP_k die Variante mit k Paaren ($|\Sigma| \geq 2$).

Für $k \leq 2$ ist PCP_k entscheidbar. Für $k \geq 7$ ist

PCP_k unentscheidbar. Für $k \in \{3, 4, 5, 6\}$ ist der Status
unbekannt.

2.8 Unentscheidbare Grammatikprobleme

Satz Gegeben 2 kontextfreie Grammatiken G_1 und G_2 , dann
sind folgende Fragestellungen unentscheidbar:

- (1) Ist $L(G_1) \cap L(G_2) = \emptyset$?
- (2) Ist $|L(G_1) \cap L(G_2)| = \infty$?
- (3) Ist $L(G_1) \subseteq L(G_2)$?
- (4) Ist $L(G_1) = L(G_2)$?
- (5) Ist $L(G_1) \cap L(G_2)$ kontextfrei?

Bew.: Durch Reduktion vom PCP auf das jeweilige Problem.

Jedem PCP $k = ((x_1, y_1), \dots, (x_k, y_k))$ über dem Alphaset $\{0, 1\}$ ordnen wir ein Paar von deterministisch kontextfreie Grammatiken G_1 und G_2 zu. Dabei sei $\Sigma = \{0, 1, \$, \underline{a_1}, \dots, \underline{a_k}\}$ stehen u.v. Parameterindizes.

G_1 :

$$S \rightarrow A \$ B$$

$$A \rightarrow a_1 A x_1 \mid a_2 A x_2 \mid \dots \mid a_k A x_k \mid a_1 x_1 \mid a_2 x_2 \mid \dots \mid a_k x_k$$

$$B \rightarrow \tilde{y}_1 B a_1 \mid \tilde{y}_2 B a_2 \mid \dots \mid \tilde{y}_k B a_k \mid \tilde{y}_1 a_1 \mid \tilde{y}_2 a_2 \mid \dots \mid \tilde{y}_k a_k$$

wobei $\tilde{w}$ das Spiegelwert von w ist.

Bsp $k = ((0, 11), (10, 0)) \quad \Sigma = \{0, 1, \$, a_1, a_2\}$

$$S \rightarrow \dots$$

$$A \rightarrow a_1 A 0 \mid a_2 A 10 \mid a_1 0 \mid a_2 10$$

$$B \rightarrow 11 B a_1 \mid 0 B a_2 \mid 11 a_1 \mid 0 a_2$$

$$\begin{array}{ccccccc} \overline{a_1} & \overline{a_2} & \overline{10} & \overline{0} & \$ & \overline{11} & \overline{a_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array}$$

G_2 :

$$S \rightarrow a_1 S a_1 \mid a_2 S a_2 \mid \dots \mid a_k S a_k \mid T$$

$$T \rightarrow 0T0 \mid 1T1 \mid \$$$

$$\overbrace{a_1 a_2 a_1 a_1 0 1 1 \$ 1 1 0 a_1 a_1 a_2 a_1}$$

$$\text{D.L. } L(G_2) = L_2 = \{ uv \$ \tilde{v} \tilde{v} \mid u \in \{a_1, \dots, a_k\}^*, v \in \{0, 1\}^* \}$$

Nun gilt:

u hat eine Lösung $i_1, \dots, i_n$

$$\begin{array}{l} L(G_1) \rightarrow \overbrace{(a_{i_1} \dots a_{i_n} \mid x_1 \dots x_n)}^{\text{falsch}} \$ \overbrace{(y_{i_1} \dots y_{i_n} \mid a_{i_1} \dots a_{i_n})} \\ L(G_2) \rightarrow \overbrace{(a_{i_1} \dots a_{i_n} \mid x_1 \dots x_n)} \$ \overbrace{(y_{i_1} \dots y_{i_n} \mid a_{i_1} \dots a_{i_n})} \end{array} \in L(G_1) \cap L(G_2)$$

Falls es so ein u und f gibt $\rightarrow$ dann hat das PCP die Lösung.

Falls das PCP eine Lösung hat, können wir das u und v berechnen. erzeugen.

D.h. $f(k) = (G_1, G_2)$ ist eine Reduktion vom DP-PCP auf Γ .

Zu (2): Wenn $L_1 \cap L_2 \neq \emptyset$, dann muss $L_1 \cap L_2$ unendlich viele Elemente enthalten, da falls k eine Lösung hat, k hat unendlich viele Lösungen.

Zu (3):

Da G_1 und G_2 def. k.f. sind kann man effektiv

G'_1 und G'_2 mit $L(G'_1) = \overline{L(G_1)}$ und $L(G'_2) = \overline{L(G_2)}$

bestimmen.

Sei G_3 Gram mit $L(G_3) = \overline{L(G_1) \cup L(G'_2)}$. G_3 ist effektiv konstruierbar (nicht notwendigerweise def. k.f.).

$$\begin{array}{lcl}
 L(G_1) \cap L(G_2) = \emptyset & \text{f.d.w.} & L(G_1) \subseteq \overline{L(G_2)} = L(G'_2) \quad (3) \checkmark \\
 & \text{g.d.w.} & L(G_1) \cup L(G'_2) = \overline{L(G_2)} \\
 & \text{g.d.w.} & L(G_3) = L(G'_2) \quad (4) \checkmark
 \end{array}$$

zu (5): Wenn $L(G_1) \cap L(G_2) = \emptyset$, dann ist $L(G_1) \cap L(\emptyset)$
kontextfrei. Wenn $L(G_1) \cap L(G_2) \neq \emptyset$, dann ist
die Sprache nicht kf., wie man mit dem Pumping-
Lemma zeigen kann. $\square$
