11a
Das Postsche Korrespondenzproblem

Prof. Dr. Jasmin Blanchette

Lehr- und Forschungseinheit für
Theoretische Informatik und Theorembeweisen

Stand: 10. Mai 2024
Basiert auf Folien von PD Dr. David Sabel

Das Postsche Korrespondenzproblem

- Das Problem wurde vorgeschlagen von Emil Post 1946.
- Es ist ein einfaches aber unentscheidbares Problem.
- Es wird häufig verwendet, um es auf andere Probleme zu reduzieren und deren Unentscheidbarkeit zu zeigen.
- Es hat nichts mit Turingmaschinen und deren Akzeptanzverhalten zu tun (im Gegensatz zu den Varianten vom Halteproblem).

Definition des Postschen Korrespondenzproblems

Definition

Gegeben sei ein Alphabet Σ und eine Folge von Wortpaaren

$$K = ((x_1, y_1), \dots, (x_k, y_k))$$

mit $x_i, y_i \in \Sigma^+$. Das Postsche Korrespondenzproblem (PCP) ist die Frage, ob es für die gegebene Folge K eine nichtleere Folge von Indizes $i_1, \dots, i_m$ mit $i_j \in \{1, \dots, k\}$ gibt, sodass

$$x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$$

Definition des Postschen Korrespondenzproblems

Definition

Gegeben sei ein Alphabet Σ und eine Folge von Wortpaaren

$$K = ((x_1, y_1), \ldots, (x_k, y_k))$$

mit $x_i, y_i \in \Sigma^+$. Das Postsche Korrespondenzproblem (PCP) ist die Frage, ob es für die gegebene Folge K eine nichtleere Folge von Indizes $i_1, \ldots, i_m$ mit $i_j \in \{1, \ldots, k\}$ gibt, sodass

$$x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$$

Als formale Sprache:

$$\begin{aligned} \mathrm{PCP} = \{ code(K) \in \Sigma^* \mid\ & \text{für } K = ((x_1, y_1), \ldots, (x_k, y_k)) \text{ gibt es eine nichtleere Folge} \\ & \text{von Indizes } i_1, \ldots, i_m \text{ mit } i_j \in \{1, \ldots, k\}, \\ & \text{sodass } x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m} \} \end{aligned}$$

PCP ist wie ein Domino-Spiel

Spielsteinarten: $\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \ldots, \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right)$.

Gesucht ist eine Aneinanderreihung der Spielsteine, sodass oben wie unten dasselbe Wort abgelesen werden kann. Dabei dürfen beliebig (aber endlich) viele Spielsteine verwendet werden.

PCP ist wie ein Domino-Spiel

Spielsteinarten: $\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \dots, \begin{bmatrix} x_k \\ y_k \end{bmatrix} \right).$

Gesucht ist eine Aneinanderreihung der Spielsteine, sodass oben wie unten dasselbe Wort abgelesen werden kann. Dabei dürfen beliebig (aber endlich) viele Spielsteine verwendet werden.

Beispiel:

Sei $K = \left(\begin{bmatrix} a \\ aba \end{bmatrix}, \begin{bmatrix} baa \\ aa \end{bmatrix}, \begin{bmatrix} ab \\ bb \end{bmatrix} \right).$

PCP ist wie ein Domino-Spiel

Spielsteinarten: $\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \ldots, \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right)$.

Gesucht ist eine Aneinanderreihung der Spielsteine, sodass oben wie unten dasselbe Wort abgelesen werden kann. Dabei dürfen beliebig (aber endlich) viele Spielsteine verwendet werden.

Beispiel:

Sei $K = \left(\begin{bmatrix} a \\ aba \end{bmatrix}, \begin{bmatrix} baa \\ aa \end{bmatrix}, \begin{bmatrix} ab \\ bb \end{bmatrix}\right)$.

$(1, 2, 3, 2)$ ist eine Lösung, da

$$\begin{bmatrix} a \\ aba \end{bmatrix}\begin{bmatrix} baa \\ aa \end{bmatrix}\begin{bmatrix} ab \\ bb \end{bmatrix}\begin{bmatrix} baa \\ aa \end{bmatrix} \begin{matrix} = abaaabbaa \\ = abaaabbaa \end{matrix}$$

Beispiel für PCP

Sei $K = \left(\begin{bmatrix} ab \\ bba \end{bmatrix}, \begin{bmatrix} ba \\ baa \end{bmatrix}, \begin{bmatrix} ba \\ aba \end{bmatrix}, \begin{bmatrix} bba \\ b \end{bmatrix} \right)$.

Beispiel für PCP

Sei $K = \left(\begin{bmatrix} ab \\ bba \end{bmatrix}, \begin{bmatrix} ba \\ baa \end{bmatrix}, \begin{bmatrix} ba \\ aba \end{bmatrix}, \begin{bmatrix} bba \\ b \end{bmatrix} \right)$.

Die kürzeste Lösung benötigt 66 Wortpaare:
(2, 1, 3, 1, 1, 2, 4, 2, 1, 3, 1, 3, 1, 1, 3, 1, 1, 2, 4, 1, 1, 2, 4, 3, 1, 4, 4, 3, 1, 1, 1, 2, 4, 2, 4, 4, 4, 3, 1, 3, 1, 4, 2, 4, 1, 1, 2, 4, 1, 4, 4, 3, 1, 4, 4, 3, 4, 4, 3, 4, 2, 4, 1, 4, 4, 3).

Unentscheidbarkeit des PCP

Der Beweis von $H \leq$ PCP erfolgt in zwei Schritten:

1. MPCP $\leq$ PCP
2. $H \leq$ MPCP.

Unentscheidbarkeit des PCP

Der Beweis von $H \leq$ PCP erfolgt in zwei Schritten:

1. MPCP $\leq$ PCP
2. $H \leq$ MPCP.

MPCP ist das Modifizierte Postsche Korrespondenzproblem:
Zulässige Lösungen müssen mit dem Index 1 beginnen.

Unentscheidbarkeit des PCP

Der Beweis von $H \leq$ PCP erfolgt in zwei Schritten:

1. MPCP $\leq$ PCP
2. $H \leq$ MPCP.

MPCP ist das Modifizierte Postsche Korrespondenzproblem:
Zulässige Lösungen müssen mit dem Index 1 beginnen.

Damit folgt aus der Unentscheidbarkeit von H die Unentscheidbarkeit von MPCP und damit die Unentscheidbarkeit des PCP.

Modifiziertes PCP

Definition

Gegeben sei ein Alphabet Σ und eine Folge von Wortpaaren

$$K = ((x_1, y_1), \ldots, (x_k, y_k))$$

mit $x_i, y_i \in \Sigma^+$. Das modifizierte Postsche Korrespondenzproblem (MPCP) ist die Frage, ob es für die gegebene Folge K eine nichtleere Folge von Indizes $\boldsymbol{i}_1 = 1, i_2, \ldots, i_m$ mit $i_j \in \{1, \ldots, k\}$ gibt, sodass

$$x_{i_1} \cdots x_{i_m} = y_{i_1} \cdots y_{i_m}$$

Reduktion von MPCP auf PCP

Lemma

MPCP $\leq$ PCP.

Reduktion von MPCP auf PCP

Lemma

MPCP $\leq$ PCP.

Beweis Gesucht ist ein berechenbares f, sodass:
K ist MPCP-lösbar g.d.w. $f(K)$ ist PCP-lösbar.

Reduktion von MPCP auf PCP

Lemma

MPCP $\leq$ PCP.

Beweis Gesucht ist ein berechenbares f, sodass:
K ist MPCP-lösbar g.d.w. $f(K)$ ist PCP-lösbar.

Für $w = a_1 \cdots a_n \in \Sigma^+$ seien

$$\bar{w} = \#a_1\#a_2\# \cdots \#a_n\# \qquad \acute{w} = a_1\#a_2\# \cdots \#a_n\# \qquad \grave{w} = \#a_1\#a_2\# \cdots \#a_n$$

Reduktion von MPCP auf PCP

Lemma

MPCP $\leq$ PCP.

Beweis Gesucht ist ein berechenbares f, sodass:
K ist MPCP-lösbar g.d.w. $f(K)$ ist PCP-lösbar.

Für $w = a_1 \cdots a_n \in \Sigma^+$ seien

$$\bar{w} = \#a_1\#a_2\# \cdots \#a_n\# \qquad \acute{w} = a_1\#a_2\# \cdots \#a_n\# \qquad \grave{w} = \#a_1\#a_2\# \cdots \#a_n$$

Sei $f\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \dots, \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right) = \left(\begin{bmatrix} \bar{x}_1 \\ \grave{y}_1 \end{bmatrix}, \begin{bmatrix} \acute{x}_1 \\ \grave{y}_1 \end{bmatrix}, \dots, \begin{bmatrix} \acute{x}_k \\ \grave{y}_k \end{bmatrix}, \begin{bmatrix} \$ \\ \#\$ \end{bmatrix}\right)$.

Beispiel für MPCP

Sei $K = \left(\begin{bmatrix} a \\ aba \end{bmatrix}, \begin{bmatrix} baa \\ aa \end{bmatrix}, \begin{bmatrix} ab \\ bb \end{bmatrix} \right)$.

Beispiel für MPCP

Sei $K = \left(\begin{bmatrix} a \\ aba \end{bmatrix}, \begin{bmatrix} baa \\ aa \end{bmatrix}, \begin{bmatrix} ab \\ bb \end{bmatrix} \right)$.

$(1, 2, 3, 2)$ ist eine MPCP-Lösung:

$$\begin{bmatrix} a \\ aba \end{bmatrix} \begin{bmatrix} baa \\ aa \end{bmatrix} \begin{bmatrix} ab \\ bb \end{bmatrix} \begin{bmatrix} baa \\ aa \end{bmatrix}$$

Beispiel für MPCP

Sei $K = \left(\begin{bmatrix} a \\ aba \end{bmatrix}, \begin{bmatrix} baa \\ aa \end{bmatrix}, \begin{bmatrix} ab \\ bb \end{bmatrix} \right)$.

$(1, 2, 3, 2)$ ist eine MPCP-Lösung:

$$\begin{bmatrix} a \\ aba \end{bmatrix} \begin{bmatrix} baa \\ aa \end{bmatrix} \begin{bmatrix} ab \\ bb \end{bmatrix} \begin{bmatrix} baa \\ aa \end{bmatrix}$$

$$f(K) = \left(\begin{bmatrix} \#a\# \\ \#a\#b\#a \end{bmatrix}, \begin{bmatrix} a\# \\ \#a\#b\#a \end{bmatrix}, \begin{bmatrix} b\#a\#a\# \\ \#a\#a \end{bmatrix}, \begin{bmatrix} a\#b\# \\ \#b\#b \end{bmatrix}, \begin{bmatrix} \$ \\ \#\$ \end{bmatrix} \right).$$

Beispiel für MPCP

Sei $K = \left(\begin{bmatrix} a \\ aba \end{bmatrix}, \begin{bmatrix} baa \\ aa \end{bmatrix}, \begin{bmatrix} ab \\ bb \end{bmatrix} \right)$.

$(1, 2, 3, 2)$ ist eine MPCP-Lösung:

$$\begin{bmatrix} a \\ aba \end{bmatrix} \begin{bmatrix} baa \\ aa \end{bmatrix} \begin{bmatrix} ab \\ bb \end{bmatrix} \begin{bmatrix} baa \\ aa \end{bmatrix}$$

$$f(K) = \left(\begin{bmatrix} \#a\# \\ \#a\#b\#a \end{bmatrix}, \begin{bmatrix} a\# \\ \#a\#b\#a \end{bmatrix}, \begin{bmatrix} b\#a\#a\# \\ \#a\#a \end{bmatrix}, \begin{bmatrix} a\#b\# \\ \#b\#b \end{bmatrix}, \begin{bmatrix} \$ \\ \#\$ \end{bmatrix} \right).$$

$(1, 3, 4, 3, 5)$ ist eine PCP-Lösung:

$$\begin{bmatrix} \#a\# \\ \#a\#b\#a \end{bmatrix} \begin{bmatrix} b\#a\#a\# \\ \#a\#a \end{bmatrix} \begin{bmatrix} a\#b\# \\ \#b\#b \end{bmatrix} \begin{bmatrix} b\#a\#a\# \\ \#a\#a \end{bmatrix} \begin{bmatrix} \$ \\ \#\$ \end{bmatrix}$$

Beispiel für MPCP

Sei $K = \left(\begin{bmatrix} a \\ aba \end{bmatrix}, \begin{bmatrix} baa \\ aa \end{bmatrix}, \begin{bmatrix} ab \\ bb \end{bmatrix} \right)$.

$(1, 2, 3, 2)$ ist eine MPCP-Lösung:

$$\begin{bmatrix} a \\ aba \end{bmatrix} \begin{bmatrix} baa \\ aa \end{bmatrix} \begin{bmatrix} ab \\ bb \end{bmatrix} \begin{bmatrix} baa \\ aa \end{bmatrix}$$

$$f(K) = \left(\begin{bmatrix} \#a\# \\ \#a\#b\#a \end{bmatrix}, \begin{bmatrix} a\# \\ \#a\#b\#a \end{bmatrix}, \begin{bmatrix} b\#a\#a\# \\ \#a\#a \end{bmatrix}, \begin{bmatrix} a\#b\# \\ \#b\#b \end{bmatrix}, \begin{bmatrix} \$ \\ \#\$ \end{bmatrix} \right).$$

$(1, 3, 4, 3, 5)$ ist eine PCP-Lösung:

$$\begin{bmatrix} \#a\# \\ \#a\#b\#a \end{bmatrix} \begin{bmatrix} b\#a\#a\# \\ \#a\#a \end{bmatrix} \begin{bmatrix} a\#b\# \\ \#b\#b \end{bmatrix} \begin{bmatrix} b\#a\#a\# \\ \#a\#a \end{bmatrix} \begin{bmatrix} \$ \\ \#\$ \end{bmatrix}$$

So ist $(1, 3, 4, 3, 5, 1, 3, 4, 3, 5)$.

Reduktion von MPCP auf PCP

Beweis Gesucht ist ein berechenbares f, sodass:
K ist MPCP-lösbar g.d.w. $f(K)$ ist PCP-lösbar.

Für $w = a_1 \cdots a_n \in \Sigma^+$ seien

$$\bar{w} = \#a_1\#a_2\# \cdots \#a_n\# \qquad \acute{w} = a_1\#a_2\# \cdots \#a_n\# \qquad \grave{w} = \#a_1\#a_2\# \cdots \#a_n$$

$$\text{Sei } f\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \dots, \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right) = \left(\begin{bmatrix} \bar{x}_1 \\ \grave{y}_1 \end{bmatrix}, \begin{bmatrix} \acute{x}_1 \\ \grave{y}_1 \end{bmatrix}, \dots, \begin{bmatrix} \acute{x}_k \\ \grave{y}_k \end{bmatrix}, \begin{bmatrix} \$ \\ \#\$ \end{bmatrix}\right).$$

Reduktion von MPCP auf PCP

Beweis Gesucht ist ein berechenbares f, sodass:
K ist MPCP-lösbar g.d.w. $f(K)$ ist PCP-lösbar.

Für $w = a_1 \cdots a_n \in \Sigma^+$ seien

$$\bar{w} = \#a_1\#a_2\#\cdots\#a_n\# \qquad \acute{w} = a_1\#a_2\#\cdots\#a_n\# \qquad \grave{w} = \#a_1\#a_2\#\cdots\#a_n$$

Sei $f\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \ldots, \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right) = \left(\begin{bmatrix} \bar{x}_1 \\ \grave{y}_1 \end{bmatrix}, \begin{bmatrix} \acute{x}_1 \\ \grave{y}_1 \end{bmatrix}, \ldots, \begin{bmatrix} \acute{x}_k \\ \grave{y}_k \end{bmatrix}, \begin{bmatrix} \$ \\ \#\$ \end{bmatrix}\right)$.

$\Longrightarrow$ Wenn $(1, i_2, \ldots, i_m)$ eine Lösung für K ist,
dann ist $(1, i_2{+}1, \ldots, i_m{+}1, k{+}2)$ eine Lösung für $f(K)$.

Reduktion von MPCP auf PCP

Beweis Gesucht ist ein berechenbares f, sodass:
K ist MPCP-lösbar g.d.w. $f(K)$ ist PCP-lösbar.

Für $w = a_1 \cdots a_n \in \Sigma^+$ seien

$$\bar{w} = \#a_1\#a_2\#\cdots\#a_n\# \qquad \acute{w} = a_1\#a_2\#\cdots\#a_n\# \qquad \grave{w} = \#a_1\#a_2\#\cdots\#a_n$$

$$\text{Sei } f\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \dots, \begin{bmatrix} x_k \\ y_k \end{bmatrix}\right) = \left(\begin{bmatrix} \bar{x}_1 \\ \grave{y}_1 \end{bmatrix}, \begin{bmatrix} \acute{x}_1 \\ \grave{y}_1 \end{bmatrix}, \dots, \begin{bmatrix} \acute{x}_k \\ \grave{y}_k \end{bmatrix}, \begin{bmatrix} \$ \\ \#\$ \end{bmatrix}\right).$$

$\Longrightarrow$ Wenn $(1, i_2, \dots, i_m)$ eine Lösung für K ist,
dann ist $(1, i_2{+}1, \dots, i_m{+}1, k{+}2)$ eine Lösung für $f(K)$.

$\Longleftarrow$ Wenn $(i_1, \dots, i_m)$ eine Lösung für $f(K)$ ist,
dann muss $i_1 = 1$ und $k + 2$ muss in der Lösung vorkommen.
Sei ℓ der kleinste Index mit $i_\ell = k + 2$.
Dann ist $(1, i_2 - 1, \dots, i_{\ell-1} - 1)$ eine Lösung für K. $\square$

Reduktion von H auf MPCP

Lemma

$H \leq$ MPCP.

Reduktion von H auf MPCP

Lemma

$H \leq$ MPCP.

Beweis Seien m eine Turingmaschinenbeschreibung und w eine Eingabe.
Gesucht ist ein berechenbares f, sodass:
die DTM M_w auf Eingabe x anhält g.d.w. $f(w\#x)$ ist MPCP-lösbar.

Reduktion von H auf MPCP

Lemma

$H \leq$ MPCP.

Beweis Seien m eine Turingmaschinenbeschreibung und w eine Eingabe.
Gesucht ist ein berechenbares f, sodass:
die DTM M_w auf Eingabe x anhält g.d.w. $f(w\#x)$ ist MPCP-lösbar.

Sei $M_w = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$.

Reduktion von H auf MPCP

Lemma

$H \leq \text{MPCP}$.

Beweis Seien m eine Turingmaschinenbeschreibung und w eine Eingabe.
Gesucht ist ein berechenbares f, sodass:
die DTM M_w auf Eingabe x anhält g.d.w. $f(w\#x)$ ist MPCP-lösbar.

Sei $M_w = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$.

Als Alphabet für das MPCP nehmen wir $\Gamma \cup Z \cup \{\#\}$.

Reduktion von H auf MPCP

Lemma

$H \leq$ MPCP.

Beweis Seien m eine Turingmaschinenbeschreibung und w eine Eingabe.
Gesucht ist ein berechenbares f, sodass:
die DTM M_w auf Eingabe x anhält g.d.w. $f(w\#x)$ ist MPCP-lösbar.

Sei $M_w = (Z, \Sigma, \Gamma, \delta, z_0, \Box, E)$.

Als Alphabet für das MPCP nehmen wir $\Gamma \cup Z \cup \{\#\}$.

Grundgedanke: Lösungen des MPCP simulieren Übergangsfolgen der DTM.

Reduktion von H auf MPCP

Lemma

$H \leq$ MPCP.

Beweis Seien m eine Turingmaschinenbeschreibung und w eine Eingabe.
Gesucht ist ein berechenbares f, sodass:
die DTM M_w auf Eingabe x anhält g.d.w. $f(w\#x)$ ist MPCP-lösbar.

Sei $M_w = (Z, \Sigma, \Gamma, \delta, z_0, \square, E)$.

Als Alphabet für das MPCP nehmen wir $\Gamma \cup Z \cup \{\#\}$.

Grundgedanke: Lösungen des MPCP simulieren Übergangsfolgen der DTM.

Das erste Wortpaar (mit dem jede Lösung anfangen muss) ist $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} \# \\ \#z_0w\# \end{bmatrix}$.

Weitere Paare lassen sich in Gruppen von „Regeln“ aufteilen:
Kopierregeln, Übergangsregeln, Löschregeln, Abschlussregeln.

Die Kopierregeln

- $\left[\begin{array}{c} a \\ a \end{array}\right]$ für alle $a \in \Gamma \cup \{\#\}$

Die Übergangsregeln

- $\left[\begin{array}{c} za \\ z'c \end{array}\right]$ falls $\delta(z,a) = (z', c, N)$

Die Übergangsregeln

- $\left[\begin{array}{c} za \\ z'c \end{array} \right]$ falls $\delta(z, a) = (z', c, N)$
- $\left[\begin{array}{c} za \\ cz' \end{array} \right]$ falls $\delta(z, a) = (z', c, R)$

Die Übergangsregeln

- $\begin{bmatrix} za \\ z'c \end{bmatrix}$ falls $\delta(z, a) = (z', c, N)$
- $\begin{bmatrix} za \\ cz' \end{bmatrix}$ falls $\delta(z, a) = (z', c, R)$
- $\begin{bmatrix} bza \\ z'bc \end{bmatrix}$ falls $\delta(z, a) = (z', c, L)$ für alle $b \in \Gamma$

Die Übergangsregeln

- $\left[\begin{array}{c} za \\ z'c \end{array}\right]$ falls $\delta(z, a) = (z', c, N)$
- $\left[\begin{array}{c} za \\ cz' \end{array}\right]$ falls $\delta(z, a) = (z', c, R)$
- $\left[\begin{array}{c} bza \\ z'bc \end{array}\right]$ falls $\delta(z, a) = (z', c, L)$ für alle $b \in \Gamma$
- $\left[\begin{array}{c} \#za \\ \#z'\Box c \end{array}\right]$ falls $\delta(z, a) = (z', c, L)$
- $\left[\begin{array}{c} z\# \\ z'c\# \end{array}\right]$ falls $\delta(z, \Box) = (z', c, N)$
- $\left[\begin{array}{c} z\# \\ cz'\# \end{array}\right]$ falls $\delta(z, \Box) = (z', c, R)$
- $\left[\begin{array}{c} bz\# \\ z'bc\# \end{array}\right]$ falls $\delta(z, \Box) = (z', c, L)$ für alle $b \in \Gamma$

Die Löschregeln

- $\begin{bmatrix} az_e \\ z_e \end{bmatrix}$ für alle $a \in \Gamma$, $z_e \in E$
- $\begin{bmatrix} z_e a \\ z_e \end{bmatrix}$ für alle $a \in \Gamma$, $z_e \in E$

Die Abschlussregeln

- $\begin{bmatrix} z_e\#\# \\ \# \end{bmatrix}$ für alle $z_e \in E$

Beispiel für die Reduktion von H auf MPCP

$$z_0abc \vdash dz_1bc \vdash dez_2c \vdash defz_3\square \vdash defz_e\square$$

Beispiel für die Reduktion von H auf MPCP

$$z_0abc \vdash dz_1bc \vdash dez_2c \vdash defz_3\square \vdash defz_e\square$$

Lösende Spielsteinfolge:

$$\begin{bmatrix}\#\\ \#z_0abc\#\end{bmatrix}\begin{bmatrix}z_0a\\ dz_1\end{bmatrix}\begin{bmatrix}b\\ b\end{bmatrix}\begin{bmatrix}c\\ c\end{bmatrix}\begin{bmatrix}\#\\ \#\end{bmatrix}\begin{bmatrix}d\\ d\end{bmatrix}\begin{bmatrix}z_1b\\ ez_2\end{bmatrix}\begin{bmatrix}c\\ c\end{bmatrix}\begin{bmatrix}\#\\ \#\end{bmatrix}\begin{bmatrix}d\\ d\end{bmatrix}\begin{bmatrix}e\\ e\end{bmatrix}\begin{bmatrix}z_2c\\ fz_3\end{bmatrix}\begin{bmatrix}\#\\ \#\end{bmatrix}\begin{bmatrix}d\\ d\end{bmatrix}\begin{bmatrix}e\\ e\end{bmatrix}\begin{bmatrix}f\\ f\end{bmatrix}\begin{bmatrix}z_3\#\\ z_e\square\#\end{bmatrix}$$

$$\begin{bmatrix}d\\ d\end{bmatrix}\begin{bmatrix}e\\ e\end{bmatrix}\begin{bmatrix}f\\ f\end{bmatrix}\begin{bmatrix}z_e\square\\ z_e\end{bmatrix}\begin{bmatrix}\#\\ \#\end{bmatrix}\begin{bmatrix}d\\ d\end{bmatrix}\begin{bmatrix}e\\ e\end{bmatrix}\begin{bmatrix}fz_e\\ z_e\end{bmatrix}\begin{bmatrix}\#\\ \#\end{bmatrix}\begin{bmatrix}d\\ d\end{bmatrix}\begin{bmatrix}ez_e\\ z_e\end{bmatrix}\begin{bmatrix}\#\\ \#\end{bmatrix}\begin{bmatrix}dz_e\\ z_e\end{bmatrix}\begin{bmatrix}\#\\ \#\end{bmatrix}\begin{bmatrix}z_e\#\#\\ \#\end{bmatrix}$$

Reduktion von H auf MPCP

Beweis (Fortsetzung)
Wir müssen zeigen:
die DTM M_w auf Eingabe x anhält g.d.w. $f(w\#x)$ genau dann MPCP-lösbar ist.

$\Longrightarrow$ Wenn M_w einen akzeptierenden Lauf hat, dann gibt es eine Folge

$$K_0 \vdash K_1 \vdash \cdots \vdash K_n$$

wobei $K_0 = z_0 w$ und $K_n = u z_e v$ für ein $z_e \in E$.

Reduktion von H auf MPCP

Beweis (Fortsetzung)

Wir müssen zeigen:

die DTM M_w auf Eingabe x anhält g.d.w. $f(w\#x)$ genau dann MPCP-lösbar ist.

$\Longrightarrow$ Wenn M_w einen akzeptierenden Lauf hat, dann gibt es eine Folge

$$K_0 \vdash K_1 \vdash \cdots \vdash K_n$$

wobei $K_0 = z_0 w$ und $K_n = u z_e v$ für ein $z_e \in E$.

Dann hat das MPCP eine Lösung, die oben und unten das Wort

$$\#K_0\#K_1\#\cdots\#K_n\#K_{n+1}\#\cdots\#K_m\#\#$$

erzeugt, wobei $K_m = z_e$ und jedes K_i mit $i \in \{n+1, \ldots, m\}$ jeweils aus K_{i-1} entsteht durch Löschen eines der benachbarten Zeichen von z_e in $u' z_e v'$ entsteht.

Reduktion von H auf MPCP

Beweis (Fortsetzung)

Die obere Folge hinkt der unteren um eine Konfiguration hinterher:

$$
\begin{array}{l}
\#K_1\#K_2\#\cdots\#K_i\# \\
\#K_1\#K_2\#\cdots\#K_i\#K_{i+1}\#
\end{array}
$$

Reduktion von H auf MPCP

Beweis (Fortsetzung)

Die obere Folge hinkt der unteren um eine Konfiguration hinterher:

$$\#K_1\#K_2\#\cdots\#K_i\#$$
$$\#K_1\#K_2\#\cdots\#K_i\#K_{i+1}\#$$

Verlängerung bis K_n:

1. Wende Kopierregeln an bis in die Nähe des Zustands.
2. Wende Übergangsregeln an.
3. Wende Kopierregeln an zum Vervollständigen

Verlängerung ab K_n:

1. Löschregeln anwenden, um die Symbole auf dem Band zu löschen.
2. Wenn in unterer Folge $z_e\#$ steht, dann Abschlussregel anwenden.

Umgekehrte Richtung

Beweis (Fortsetzung)

$\Longleftarrow$ Jede Lösung für das MPCP (welches ja mit dem ersten Spielstein beginnen muss) erzeugt einen akzeptierende Lauf, der bezeugt, dass die Turingmaschine bei Eingabe w hält.

Umgekehrte Richtung

Beweis (Fortsetzung)

$\Longleftarrow$ Jede Lösung für das MPCP (welches ja mit dem ersten Spielstein beginnen muss) erzeugt einen akzeptierende Lauf, der bezeugt, dass die Turingmaschine bei Eingabe w hält.

Wegen der Kopienregeln können MPCP-Lösungen Wiederholungen von der Form $\#K\#K\#$ enthalten. Diese können vereinfacht werden, um einen Lauf zu erhalten.

Umgekehrte Richtung

Beweis (Fortsetzung)

$\Longleftarrow$ Jede Lösung für das MPCP (welches ja mit dem ersten Spielstein beginnen muss) erzeugt einen akzeptierende Lauf, der bezeugt, dass die Turingmaschine bei Eingabe w hält.

Wegen der Kopienregeln können MPCP-Lösungen Wiederholungen von der Form $\#K\#K\#$ enthalten. Diese können vereinfacht werden, um einen Lauf zu erhalten.

Eine MPCP-Lösung kann auch mehreren Läufen hintereinander entsprechen. Dann betrachten wir nur den ersten Lauf.

Umgekehrte Richtung

Beweis (Fortsetzung)

$\Longleftarrow$ Jede Lösung für das MPCP (welches ja mit dem ersten Spielstein beginnen muss) erzeugt einen akzeptierende Lauf, der bezeugt, dass die Turingmaschine bei Eingabe w hält.

Wegen der Kopienregeln können MPCP-Lösungen Wiederholungen von der Form $\#K\#K\#$ enthalten. Diese können vereinfacht werden, um einen Lauf zu erhalten.

Eine MPCP-Lösung kann auch mehreren Läufen hintereinander entsprechen. Dann betrachten wir nur den ersten Lauf.

Schließlich prüfen wir, dass f berechenbar ist. $\square$

Unentscheidbarkeit PCP und MPCP

Satz

Das Postsche Korrespondenzproblem (sowie das modifizierte Postsche Korrespondenzproblem) ist unentscheidbar.

Unentscheidbarkeit PCP und MPCP

Satz

Das Postsche Korrespondenzproblem (sowie das modifizierte Postsche Korrespondenzproblem) ist unentscheidbar.

Beweis Da H unentscheidbar ist und $H \leq \text{MPCP} \leq \text{PCP}$ gilt, folgt, dass MPCP und PCP unentscheidbar sind. $\square$

PCP für binäre Alphabete

Lemma

Das Postsche Korrespondenzproblem über dem Alphabet Σ mit $|\Sigma| = 2$ (01-PCP) ist unentscheidbar.

PCP für binäre Alphabete

Lemma

Das Postsche Korrespondenzproblem über dem Alphabet Σ mit $|\Sigma| = 2$ (01-PCP) ist unentscheidbar.

Beweis Wir reduzieren PCP auf 01-PCP.

PCP für binäre Alphabete

Lemma

Das Postsche Korrespondenzproblem über dem Alphabet Σ mit $|\Sigma| = 2$ (01-PCP) ist unentscheidbar.

Beweis Wir reduzieren PCP auf 01-PCP.

Sei $K = (x_1, y_1), \ldots, (x_k, y_k)$ eine Instanz des PCP über dem Alphabet $\{a_1, \ldots, a_j\}$.

PCP für binäre Alphabete

Lemma

Das Postsche Korrespondenzproblem über dem Alphabet Σ mit $|\Sigma| = 2$ (01-PCP) ist unentscheidbar.

Beweis Wir reduzieren PCP auf 01-PCP.

Sei $K = (x_1, y_1), \ldots, (x_k, y_k)$ eine Instanz des PCP über dem Alphabet $\{a_1, \ldots, a_j\}$.

O.B.d.A. sei $\Sigma = \{0, 1\}$.

PCP für binäre Alphabete

Lemma

Das Postsche Korrespondenzproblem über dem Alphabet Σ mit $|\Sigma| = 2$ (01-PCP) ist unentscheidbar.

Beweis Wir reduzieren PCP auf 01-PCP.

Sei $K = (x_1, y_1), \ldots, (x_k, y_k)$ eine Instanz des PCP über dem Alphabet $\{a_1, \ldots, a_j\}$.

O.B.d.A. sei $\Sigma = \{0, 1\}$.

Sei $f(a_i) = 10^i$,
$f(\varepsilon) = \varepsilon$, $f(a_i w) = f(a_i) f(w)$ und
$f(K) = (f(x_1), f(y_1)), \ldots, (f(x_k), f(y_k))$.

PCP für binäre Alphabete

Lemma

Das Postsche Korrespondenzproblem über dem Alphabet Σ mit $|\Sigma| = 2$ (01-PCP) ist unentscheidbar.

Beweis Wir reduzieren PCP auf 01-PCP.

Sei $K = (x_1, y_1), \ldots, (x_k, y_k)$ eine Instanz des PCP über dem Alphabet $\{a_1, \ldots, a_j\}$.

O.B.d.A. sei $\Sigma = \{0, 1\}$.

Sei $f(a_i) = 10^i$,
$f(\varepsilon) = \varepsilon$, $f(a_i w) = f(a_i) f(w)$ und
$f(K) = (f(x_1), f(y_1)), \ldots, (f(x_k), f(y_k))$.

Dann gilt: $i_1, \ldots, i_n$ ist eine PCP-Lösung für K g.d.w. $i_1, \ldots, i_n$ ist eine 01-PCP-Lösung für $f(K)$.

PCP für binäre Alphabete

Lemma

Das Postsche Korrespondenzproblem über dem Alphabet Σ mit $|\Sigma| = 2$ (01-PCP) ist unentscheidbar.

Beweis Wir reduzieren PCP auf 01-PCP.

Sei $K = (x_1, y_1), \ldots, (x_k, y_k)$ eine Instanz des PCP über dem Alphabet $\{a_1, \ldots, a_j\}$.

O.B.d.A. sei $\Sigma = \{0, 1\}$.

Sei $f(a_i) = 10^i$,
$f(\varepsilon) = \varepsilon$, $f(a_i w) = f(a_i)f(w)$ und
$f(K) = (f(x_1), f(y_1)), \ldots, (f(x_k), f(y_k))$.

Dann gilt: $i_1, \ldots, i_n$ ist eine PCP-Lösung für K g.d.w. $i_1, \ldots, i_n$ ist eine 01-PCP-Lösung für $f(K)$.

Schließlich ist f turingberechenbar. $\square$

Beispiel für 01-PCP

Sei $K = \left(\begin{bmatrix} a_2 a_1 a_1 \\ a_1 a_1 \end{bmatrix}, \begin{bmatrix} a_3 \\ a_3 a_2 a_1 \end{bmatrix}, \begin{bmatrix} a_1 a_2 \\ a_2 a_2 \end{bmatrix} \right)$

Beispiel für 01-PCP

Sei $K = \left(\begin{bmatrix} a_2 a_1 a_1 \\ a_1 a_1 \end{bmatrix}, \begin{bmatrix} a_3 \\ a_3 a_2 a_1 \end{bmatrix}, \begin{bmatrix} a_1 a_2 \\ a_2 a_2 \end{bmatrix} \right)$

$(2, 1, 3, 1)$ ist eine PCP-Lösung:

$$\begin{bmatrix} a_3 \\ a_3 a_2 a_1 \end{bmatrix} \begin{bmatrix} a_2 a_1 a_1 \\ a_1 a_1 \end{bmatrix} \begin{bmatrix} a_1 a_2 \\ a_2 a_2 \end{bmatrix} \begin{bmatrix} a_2 a_1 a_1 \\ a_1 a_1 \end{bmatrix}$$

Beispiel für 01-PCP

Sei $K = \left(\begin{bmatrix} a_2 a_1 a_1 \\ a_1 a_1 \end{bmatrix}, \begin{bmatrix} a_3 \\ a_3 a_2 a_1 \end{bmatrix}, \begin{bmatrix} a_1 a_2 \\ a_2 a_2 \end{bmatrix} \right)$

$(2, 1, 3, 1)$ ist eine PCP-Lösung:

$$\begin{bmatrix} a_3 \\ a_3 a_2 a_1 \end{bmatrix} \begin{bmatrix} a_2 a_1 a_1 \\ a_1 a_1 \end{bmatrix} \begin{bmatrix} a_1 a_2 \\ a_2 a_2 \end{bmatrix} \begin{bmatrix} a_2 a_1 a_1 \\ a_1 a_1 \end{bmatrix}$$

$$f(K) = \left(\begin{bmatrix} 1001010 \\ 1010 \end{bmatrix}, \begin{bmatrix} 1000 \\ 100010010 \end{bmatrix}, \begin{bmatrix} 10100 \\ 100100 \end{bmatrix} \right)$$

Beispiel für 01-PCP

Sei $K = \left(\begin{bmatrix} a_2 a_1 a_1 \\ a_1 a_1 \end{bmatrix}, \begin{bmatrix} a_3 \\ a_3 a_2 a_1 \end{bmatrix}, \begin{bmatrix} a_1 a_2 \\ a_2 a_2 \end{bmatrix} \right)$

$(2, 1, 3, 1)$ ist eine PCP-Lösung:

$$\begin{bmatrix} a_3 \\ a_3 a_2 a_1 \end{bmatrix} \begin{bmatrix} a_2 a_1 a_1 \\ a_1 a_1 \end{bmatrix} \begin{bmatrix} a_1 a_2 \\ a_2 a_2 \end{bmatrix} \begin{bmatrix} a_2 a_1 a_1 \\ a_1 a_1 \end{bmatrix}$$

$f(K) = \left(\begin{bmatrix} 1001010 \\ 1010 \end{bmatrix}, \begin{bmatrix} 1000 \\ 100010010 \end{bmatrix}, \begin{bmatrix} 10100 \\ 100100 \end{bmatrix} \right)$

$(2, 1, 3, 1)$ ist eine 01-PCP-Lösung:

$$\begin{bmatrix} 1000 \\ 100010010 \end{bmatrix} \begin{bmatrix} 1001010 \\ 1010 \end{bmatrix} \begin{bmatrix} 10100 \\ 100100 \end{bmatrix} \begin{bmatrix} 1001010 \\ 1010 \end{bmatrix}$$

PCP für unäre Alphabete

Lemma

Das PCP für unäre Alphabete ist entscheidbar.

PCP für unäre Alphabete

Lemma

Das PCP für unäre Alphabete ist entscheidbar.

Beweis Alle Wortpaare sind von der Form $\begin{bmatrix} a^n \\ a^m \end{bmatrix}$.

PCP für unäre Alphabete

Lemma

Das PCP für unäre Alphabete ist entscheidbar.

Beweis Alle Wortpaare sind von der Form $\begin{bmatrix} a^n \\ a^m \end{bmatrix}$.

Wenn $|x_i| < |y_i|$ für alle (x_i, y_i) gilt, dann gibt es keine Lösung.

PCP für unäre Alphabete

Lemma

Das PCP für unäre Alphabete ist entscheidbar.

Beweis Alle Wortpaare sind von der Form $\left[\begin{array}{c} a^n \\ a^m \end{array}\right]$.

Wenn $|x_i| < |y_i|$ für alle (x_i, y_i) gilt, dann gibt es keine Lösung.

Wenn $|x_i| > |y_i|$ für alle (x_i, y_i) gilt, dann gibt es keine Lösung.

PCP für unäre Alphabete

Lemma

Das PCP für unäre Alphabete ist entscheidbar.

Beweis Alle Wortpaare sind von der Form $\left[\begin{array}{c} a^n \\ a^m \end{array}\right]$.

Wenn $|x_i| < |y_i|$ für alle (x_i, y_i) gilt, dann gibt es keine Lösung.

Wenn $|x_i| > |y_i|$ für alle (x_i, y_i) gilt, dann gibt es keine Lösung.

Wenn $(x_i, y_i) = (a^n, a^{n+r})$ und $(x_j, y_j) = (a^{m+s}, a^m)$ mit $s, r \geq 0$,
dann ist das PCP immer lösbar.
Die Lösung ist $\underbrace{i, \ldots, i}_{s\text{-mal}}, \underbrace{j, \ldots, j}_{r\text{-mal}}$, denn oben $a^{s \cdot n + r \cdot (m+s)}$ und unten $a^{s \cdot (n+r) + r \cdot m}$.

Daher oben wie unten $sn + rm + rs$ viele a's. □

Beispiel für PCP mit unärem Alphabet

Sei $K = \left(\begin{bmatrix} a \\ aaaa \end{bmatrix}, \begin{bmatrix} aaa \\ a \end{bmatrix} \right)$

Beispiel für PCP mit unärem Alphabet

Sei $K = \left(\begin{bmatrix} a \\ aaaa \end{bmatrix}, \begin{bmatrix} aaa \\ a \end{bmatrix} \right)$

$(1, 1, 2, 2, 2)$ ist eine Lösung:

$$\begin{bmatrix} a \\ aaaa \end{bmatrix} \begin{bmatrix} a \\ aaaa \end{bmatrix} \begin{bmatrix} aaa \\ a \end{bmatrix} \begin{bmatrix} aaa \\ a \end{bmatrix} \begin{bmatrix} aaa \\ a \end{bmatrix}$$

Anzahl k der Spielsteinarten beschränken

PCP mit k vielen verschiedenen Spielsteinarten:

- $k = 1$ und $k = 2$: als entscheidbar gezeigt 1982
- $k \geq 5$: als unentscheidbar gezeigt 2015
- $k = 3$ und $k = 4$: unbekannt.

Semi-Entscheidbarkeit des PCP

PCP ist semi-entscheidbar:

1. Probiere alle Folgen von i Wortpaaren aus.
2. Lasse i wachsen.

Diese Prozedur findet eine Lösung, wenn eine existiert, in endlich vielen Schritten, aber terminiert nicht, wenn keine Lösung existiert.

Universelle Turingmaschine

Da $H \leq \mathrm{PCP}$ folgt auch, dass H semi-entscheidbar ist.

Universelle Turingmaschine

Da $H \leq \mathrm{PCP}$ folgt auch, dass H semi-entscheidbar ist.

D.h. es gibt eine DTM, die sich bei Eingabe $w\#x$ so verhält wie M_w auf Eingabe x was das Halten betrifft.

Universelle Turingmaschine

Da $H \leq$ PCP folgt auch, dass H semi-entscheidbar ist.

D.h. es gibt eine DTM, die sich bei Eingabe $w\#x$ so verhält wie M_w auf Eingabe x was das Halten betrifft.

Ferner: Es gibt eine DTM U, die sich bei Eingabe $w\#x$ so verhält wie M_w auf Eingabe x.

Universelle Turingmaschine

Da $H \leq \text{PCP}$ folgt auch, dass H semi-entscheidbar ist.

D.h. es gibt eine DTM, die sich bei Eingabe $w\#x$ so verhält wie M_w auf Eingabe x was das Halten betrifft.

Ferner: Es gibt eine DTM U, die sich bei Eingabe $w\#x$ so verhält wie M_w auf Eingabe x.

U nennt man eine universelle Turingmaschine:

- U verhält sich wie ein Interpreter für Turingmaschinen.
- U wird durch die Eingabe w programmiert und x ist dann die eigentliche Eingabe für das Programm.