

Theory of Computer Science

C5. Post Correspondence Problem

Gabriele Röger

University of Basel

April 28, 2025

1 / 27

Theory of Computer Science

April 28, 2025 — C5. Post Correspondence Problem

C5.1 Post Correspondence Problem

C5.2 (Un-)Decidability of PCP

C5.3 Summary

2 / 27

C5. Post Correspondence Problem

Post Correspondence Problem

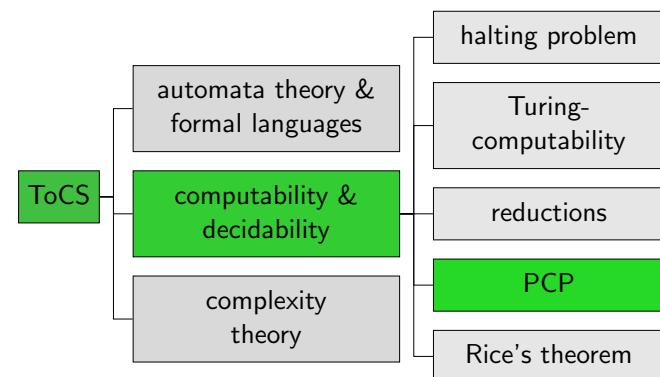
C5.1 Post Correspondence Problem

3 / 27

C5. Post Correspondence Problem

Post Correspondence Problem

Content of the Course



4 / 27

More Options for Reduction Proofs?

- ▶ We can prove the undecidability of a problem with a reduction from an undecidable problem.
- ▶ The [halting problem](#) and the [halting problem on the empty tape](#) are possible options for this.
- ▶ both halting problem variants are quite similar 😊

→ We want a wider selection for reduction proofs
 → Is there some problem that is different in flavor?

Post correspondence problem

(named after mathematician [Emil Leon Post](#))

Post Correspondence Problem: Example

Example (Post Correspondence Problem)

Given: different kinds of "dominos"

1: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr><tr><td>101</td></tr></table>	1	101	2: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>10</td></tr><tr><td>00</td></tr></table>	10	00	3: <table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>011</td></tr><tr><td>11</td></tr></table>	011	11
1								
101								
10								
00								
011								
11								

(an infinite number of each kind)

Question: Is there a sequence of dominos such that
 the upper and lower row match (= are equal)

<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>1</td></tr><tr><td>101</td></tr></table>	1	101	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>011</td></tr><tr><td>11</td></tr></table>	011	11	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>10</td></tr><tr><td>00</td></tr></table>	10	00	<table border="1" style="display: inline-table; vertical-align: middle;"><tr><td>011</td></tr><tr><td>11</td></tr></table>	011	11
1											
101											
011											
11											
10											
00											
011											
11											
1	3	2	3								

Post Correspondence Problem: Definition

Definition (Post Correspondence Problem PCP)

Given: Finite sequence of pairs of words
 $(t_1, b_1), (t_2, b_2), \dots, (t_k, b_k)$, where $t_i, b_i \in \Sigma^+$
 (for an arbitrary alphabet Σ)

Question: Is there a sequence
 $i_1, i_2, \dots, i_n \in \{1, \dots, k\}$, $n \geq 1$,
 with $t_{i_1} t_{i_2} \dots t_{i_n} = b_{i_1} b_{i_2} \dots b_{i_n}$?

A [solution](#) of the correspondence problem is such a sequence
 i_1, \dots, i_n , which we call a [match](#).

Exercise (slido)

Consider PCP instance $(11, 1), (0, 00), (10, 01), (01, 11)$.



Is 2, 4, 3, 3, 1 a match?

Given-Question Form vs. Definition as Set

So far: problems defined as sets

Now: definition in **Given-Question form**

Definition (new problem P)

Given: Instance \mathcal{I}

Question: Does \mathcal{I} have a specific property?

corresponds to definitions

Definition (new problem P)

The problem P is the language

$P = \{w \mid w \text{ encodes an instance } \mathcal{I} \text{ with the required property}\}$.

Definition (new problem P)

The problem P is the language

$P = \{\langle\langle \mathcal{I} \rangle\rangle \mid \mathcal{I} \text{ is an instance with the required property}\}$.

PCP Definition as Set

We can alternatively define PCP as follows:

Definition (Post Correspondence Problem PCP)

The Post Correspondence Problem PCP is the set

$\text{PCP} = \{w \mid w \text{ encodes a sequence of pairs of words}$

$(t_1, b_1), (t_2, b_2), \dots, (t_k, b_k)$, for which
there is a sequence $i_1, i_2, \dots, i_n \in \{1, \dots, k\}$
such that $t_{i_1} t_{i_2} \dots t_{i_n} = b_{i_1} b_{i_2} \dots b_{i_n}\}$.

C5.2 (Un-)Decidability of PCP

Post Correspondence Problem

PCP cannot be so hard, huh?

– Is it?

1101	0110	1
1	11	110

Formally: $K = ((1101, 1), (0110, 11), (1, 110))$
→ Shortest match has length 252!

10	0	100
0	001	1

Formally: $K = ((10, 0), (0, 001), (100, 1))$
→ Unsolvable

PCP: Turing-recognizability

Theorem (Turing-recognizability of PCP)

PCP is *Turing-recognizable*.

Proof.

Recognition procedure for input w :

- ▶ If w encodes a sequence $(t_1, b_1), \dots, (t_k, b_k)$ of pairs of words:
Test systematically longer and longer sequences i_1, i_2, \dots, i_n
whether they represent a match.
If yes, terminate and return "yes".
- ▶ If w does not encode such a sequence: enter an infinite loop.

If $w \in \text{PCP}$ then the procedure terminates with "yes",
otherwise it does not terminate. □

MPCP: Definition

Definition (Modified Post Correspondence Problem MPCP)

Given: Sequence of word pairs as for PCP

Question: Is there a match $i_1, i_2, \dots, i_n \in \{1, \dots, k\}$
with $i_1 = 1$?

PCP: Undecidability

Theorem (Undecidability of PCP)

PCP is *undecidable*.

Proof via an intermediate other problem
modified PCP (MPCP)

- ① Reduce MPCP to PCP ($\text{MPCP} \leq \text{PCP}$)
- ② Reduce halting problem to MPCP ($H \leq \text{MPCP}$)

→ Let's get started...

Reducibility of MPCP to PCP(1)

Lemma

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

Proof.

Let $\#, \$ \notin \Sigma$. For word $w = a_1 a_2 \dots a_m \in \Sigma^+$ define

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

For input $C = ((t_1, b_1), \dots, (t_k, b_k))$ define
 $f(C) = ((\bar{t}_1, \grave{b}_1), (\acute{t}_1, b_1), (\acute{t}_2, \grave{b}_2), \dots, (\acute{t}_k, \grave{b}_k), (\$, \#\$))$

...

Reducibility of MPCP to PCP(2)

Proof (continued).

$$f(C) = ((\bar{t}_1, \bar{b}_1), (\acute{t}_1, \acute{b}_1), (\acute{t}_2, \acute{b}_2), \dots, (\acute{t}_k, \acute{b}_k), (\$, \#\#))$$

Function f is [computable](#), and can suitably get extended to a [total](#) function. It holds that

C has a solution with $i_1 = 1$ iff $f(C)$ has a solution:

Let $1, i_2, i_3, \dots, i_n$ be a solution for C . Then

$1, i_2 + 1, \dots, i_n + 1, k + 2$ is a solution for $f(C)$.

If i_1, \dots, i_n is a match for $f(C)$, then (due to the construction of the word pairs) there is a $m \leq n$ such that $i_1 = 1, i_m = k + 2$ and $i_j \in \{2, \dots, k + 1\}$ for $j \in \{2, \dots, m - 1\}$. Then

$1, i_2 - 1, \dots, i_{m-1} - 1$ is a solution for C .

$\Rightarrow f$ is a reduction from MPCP to PCP. □

Reducibility of H to MPCP(1)

Lemma

$$H \leq \text{MPCP}.$$

Proof.

Goal: Construct for Turing machine

$M = \langle Q, \Sigma, \Gamma, \delta, q_0, q_{\text{accept}}, q_{\text{reject}} \rangle$ and word $w \in \Sigma^*$ an MPCP instance $C = ((t_1, b_1), \dots, (t_k, b_k))$ such that

M started on w terminates iff $C \in \text{MPCP}$.

...

PCP: Undecidability – Where are we?

Theorem (Undecidability of PCP)

PCP is [undecidable](#).

Proof via an intermediate other problem

[modified PCP \(MPCP\)](#)

- ➊ Reduce MPCP to PCP ($\text{MPCP} \leq \text{PCP}$) ✓
- ➋ Reduce halting problem to MPCP ($H \leq \text{MPCP}$)

Reducibility of H to MPCP(2)

Proof (continued).

Idea:

- ▶ Sequence of words describes sequence of configurations of the TM
- ▶ “ t -row” follows “ b -row” $x : [\# \ c_0 \ \# \ c_1 \ \# \ c_2 \ \#]$
 $y : [\# \ c_0 \ \# \ c_1 \ \# \ c_2 \ \# \ c_3 \ \#]$
- ▶ Configurations get mostly just copied, only the area around the head changes.
- ▶ After a terminating configuration has been reached: make row equal by deleting the configuration.

...

Reducibility of H to MPCP(3)

Proof (continued).

Alphabet of C is $\Gamma \cup Q \cup \{\#\}$.

1. Pair: $(\#, \#q_0 w \#)$

Other pairs:

① copy: (a, a) for all $a \in \Gamma \cup \{\#\}$

② transition:

(qa, cq') if $\delta(q, a) = (q', c, R)$

$(q\#, cq'\#)$ if $\delta(q, \square) = (q', c, R)$

...

21 / 27

Reducibility of H to MPCP(4)

Proof (continued).

$(bqa, q'bc)$ if $\delta(q, a) = (q', c, L)$ for all $b \in \Gamma$

$(bq\#, q'bc\#)$ if $\delta(q, \square) = (q', c, L)$ for all $b \in \Gamma$

$(\#qa, \#q'c)$ if $\delta(q, a) = (q', c, L)$

$(\#q\#, \#q'c\#)$ if $\delta(q, \square) = (q', c, L)$

③ deletion: (aq, q) and (qa, q)

for all $a \in \Gamma$ and $q \in \{q_{\text{accept}}, q_{\text{reject}}\}$

④ finish: $(q\#\#, \#)$ for all $q \in \{q_{\text{accept}}, q_{\text{reject}}\}$

...

22 / 27

Reducibility of H to MPCP(5)

Proof (continued).

“ \Rightarrow ” If M terminates on input w , there is a sequence c_0, \dots, c_t of configurations with

- ▶ $c_0 = q_0 w$ is the start configuration
- ▶ c_t is a terminating configuration
- ▶ $(c_t = uq_e v \text{ mit } u, v \in \Gamma^* \text{ and } q_e \in \{q_{\text{accept}}, q_{\text{reject}}\})$
- ▶ $c_i \vdash c_{i+1}$ for $i = 0, 1, \dots, t-1$

Then C has a match with the overall word

$\#c_0\#c_1\#\dots\#c_t\#c'_t\#c''_t\#\dots\#q_e\#\#$

Up to c_t : “ t -row” follows “ b -row”

From c'_t : deletion of symbols adjacent to terminating state. ...

23 / 27

Reducibility of H to MPCP(6)

Proof (continued).

“ \Leftarrow ” If C has a solution, it has the form

$\#c_0\#c_1\#\dots\#c_n\#\#$,

with $c_0 = q_0 w$. Moreover, there is an $\ell \leq n$, such that q_{accept} or q_{reject} occurs for the first time in c_ℓ .

All c_i for $i \leq \ell$ are configurations of M and $c_i \vdash c_{i+1}$ for $i \in \{0, \dots, \ell-1\}$.

c_0, \dots, c_ℓ is hence the sequence of configurations of M on input w , which shows that the TM terminates. \square

24 / 27

PCP: Undecidability – Done!

Theorem (Undecidability of PCP)

PCP is *undecidable*.

Proof via an intermediate other problem

modified PCP (MPCP)

- ➊ Reduce MPCP to PCP ($\text{MPCP} \leq \text{PCP}$) ✓
- ➋ Reduce halting problem to MPCP ($H \leq \text{MPCP}$) ✓

Proof.

Due to $H \leq \text{MPCP}$ and $\text{MPCP} \leq \text{PCP}$ it holds that $H \leq \text{PCP}$.

Since H is undecidable, also PCP must be undecidable. \square

C5.3 Summary

Summary

► Post Correspondence Problem:

Find a sequence of word pairs s.t. the concatenation of all first components equals the one of all second components.

► The Post Correspondence Problem is **Turing-recognizable** but **not decidable**.