HAL
open science

On Bi-infinite and Conjugate Post Correspondence Problems

Olivier Finkel, Vesa Halava, Tero Harju, Esa Sahla

To cite this version:

Olivier Finkel, Vesa Halava, Tero Harju, Esa Sahla. On Bi-infinite and Conjugate Post Correspondence Problems. RAIRO - Theoretical Informatics and Applications (RAIRO: ITA), 2023, 57, Article number 7, p. 1-11. 10.1051/ita/2023008 . hal-03780128

HAL Id: hal-03780128

https://hal.science/hal-03780128v1

Submitted on 19 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

On Bi-infinite and Conjugate Post Correspondence Problems

Olivier Finkel[1], Vesa Halava*[2], Tero Harju[2], and Esa Sahla[2]

[1]Institut de Mathématiques de Jussieu - Paris Rive Gauche, CNRS, Université Paris Cité, Sorbonne Université, Paris, France.
finkel@math.univ-paris-diderot.fr
[2]Department of Mathematics and Statistics, University of Turku, Finland. *{vesa.halava,harju,esa.sahla}@utu.fi*

September 10, 2022

Abstract

We study two modifications of the Post Correspondence Problem (PCP), namely 1) the bi-infinite version, where it is asked whether there exists a bi-infinite word such that two given morphisms agree on it, and 2) the conjugate version, where we require the images of a solution for two given morphisms are conjugates of each other. For the bi-infinite PCP we show that it is in the class Σ_2^0 of the arithmetical hierarchy and for the conjugate PCP we give an undecidability proof by reducing it to the word problem for a special type of semi-Thue systems.

Keywords: Bi-infinite words, Conjugate words, Post Correspondence Problem, Undecidability

1 Introduction

The original formulation of the *Post Correspondence Problem* (PCP) by Emil Post in [16] is the following:

*Supported by emmy.network foundation under the aegis of the Fondation de Luxembourg.

Problem 1 (PCP)**.** *Let A be a finite alphabet. Given a finite set of pairs of words over A, say $(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)$, does there exists a nonempty sequence $i_1, \ldots, i_k$ of indices such that*

$$u_{i_1} u_{i_2} \cdots u_{i_k} = v_{i_1} v_{i_2} \cdots v_{i_k} \,?$$

Post proved that the PCP is undecidable in [16]. Since then the PCP and its many variants have been used as a bridge from combinatorial undecidable problems of computational systems and formal rewriting systems to decision problems in algebraic settings. The PCP is usually defined as a problem in free word monoids (as A^* is the free monoid of all finite words over A with catenation as the operation). Indeed, the PCP is equivalent to asking for two given morphisms $g, h\colon B^* \to A^*$, whether or not there exists a non-empty word w such that

$$g(w) = h(w).$$

Note that we may choose set $B = \{1, \ldots, n\}$, and $g(i) = u_i$, $h(i) = v_i$ for all $i = 1, \ldots, n$ where (u_i, v_i) is a pair in the original formulation of the PCP. The variants of the PCP also reveal the boundary between decidability and undecidability. It is known that the PCP is decidable for $n = 2$, see [4, 10], and undecidable for $n = 5$, see [14]. On the other hand, it is known that the infinite PCP, asking whether there is a (right) infinite sequence $i_1, i_2, \ldots$ of the indices such that the words agree, is decidable for two pairs of words, see [11], and undecidable for 8 pairs, see [3]. It has been proved that the infinite PCP is not "more complex" than the PCP with respect to the arithmetical hierarchy, see [7], where it was proved that the infinite PCP is Π^0_1-complete as the PCP is known to be Σ^0_1-complete.

In this paper we study two variants of the PCP. The first variant is called the *bi-infinite Post Correspondence Problem* ($\mathbb{Z}$PCP), where it is asked whether or not there exits a bi-infinite sequence of the indices such that the words agree. The morphisms version of the $\mathbb{Z}$PCP is the following:

Problem 2. *Given two morphisms $h, g\colon A^* \to B^*$, does the exist a bi-infinite word w such that $h(w) = g(w)$.*

Note that already the equality of the images of bi-infinite words needs to be defined properly: for a bi-infinite word w, $h(w) = g(w)$ if and only if there is a constant $s \in \mathbb{Z}$ such that for all letters $h(w)(i) = g(w)(i + s)$ for all positions $i \in \mathbb{Z}$. An *instance* of the $\mathbb{Z}$ PCP is a pair of morphisms (h, g) and a bi-infinite word w satisfying $h(w) = g(w)$ is said to be a *solution* of the instance (h, g).

Our second variant deals with conjugate words. Two words x and y are *conjugates* if there exist words u and v such that $x = uv$ and $y = vu$.

We call the following problem the *conjugate-PCP*.

Problem 3. *Given two morphisms $h, g : A^* \to B^*$, does there exist a word $w \in A^+$ such that $h(w) = uv$ and $g(w) = vu$ for some words $u, v \in B^*$.*

The behaviour of the instances of the conjugate-PCP differ vastly from the more traditional variants where a valid presolution (prefix of a candidate solution) can be verified by aligning the matching parts of the images. Working with the possible solutions of the instances of the conjugate-PCP is less intuitive.

For example let us have morphisms h, g and we guess that a solution w begins with the letter a. Then the situation is the following:

$$\begin{aligned} h(w) &= \underbrace{h(a) \quad \cdots \quad \underbrace{g(a) \quad \cdots}_{v}}_{u} \\ g(w) &= \underbrace{g(a) \quad \cdots \quad \underbrace{h(a) \quad \cdots}_{u}}_{v} \end{aligned}$$

The validity of the presolution a cannot be verified because there may not be any matching between $h(a)$ and $g(a)$. Moreover the factorization of the images to u and v need not be unique even for minimal solutions:

Example. Let $h, g : \{a, b\}^* \to \{a, b\}^*$ be morphisms defined by

$$\begin{aligned} h(a) &= aba, & g(a) &= bab, \\ h(b) &= b, & g(b) &= a. \end{aligned}$$

Now ab is a minimal solution for the conjugate-PCP instance (h, g) having two factorizations: $u = a, v = bab$ or $u = aba, v = b$.

Both variants defined above were originally proved to be undecidable in [18] using linearly bounded automata (LBA)[1]. Indeed, the conjugate-PCP was not directly proved in [18], although it is claimed so in [19]. Let us consider the terminology of Ruohonen in [19] in a bit more details: Let u and v be words, and denote $u \sim_m v$ if there exist words $u_1, \ldots, u_m$ and a permutation τ on the set $\{1, \ldots, m\}$ such that $u = u_1 \cdots u_m$ and $v = u_{\tau(1)} \cdots u_{\tau(m)}$.

[1] Note that in [18] the $\mathbb{Z}$PCP is called *doubly infinite PCP*

Problem 4 ((m,n)-permutation PCP)**.** *Given morphisms $g, h\colon A^* \to B^*$ does there exists words $u, v \in A^*$ such that*

$$u \sim_m v \text{ and } g(u) \sim_n h(v).$$

Obviously, our formulation of the conjugate-PCP is the (1,2)-permutational PCP of Ruohonen. Now in [19], it is mentioned that (1,2)-permutational PCP was proved to be undecidable in [18], but the problem is not explicitly mentioned there. On the other hand, the (2,2)-permutational PCP is shown to be undecidable in [18][2]. It is possible that Ruohonen uses the later result without details, because of the following simple lemma, which follows from special cyclic shift property of permutations.

Lemma 1. *For morphisms $g, h :: A^* \to B^*$, the instance (g, h) has a solution to the $(1, 2)$-permutational PCP if and only if it has a solution to the $(2, 2)$-permutational PCP.*

Proof. Firstly, a solution to the $(1, 2)$-permutational PCP is a solution to the $(2, 2)$-permutational PCP where the first permutation on the pre-image being trivial.

Secondly, assume that there exists a solution $u = xy, v = yx$, two $(2, 2)$-permutational PCP. So $g(xy) = zw$ and $h(yx) = wz$ for some $w, z \in B^*$. We have two cases, either $h(y)$ is a prefix of w, or vice versa.

In the first case, $w = h(y)r$ for some word $r \in B^*$, and $h(xy) = rzh(y)$ and $g(xy) = zw = zh(y)r$ and, therefore, xy is a solution for the $(1, 2)$-permutational PCP.

In the second case, $h(y) = wr$, $z = rh(x)$, for some word $r \in B^*$. Then $g(xy) = rh(x)w$ and $h(xy) = h(x)wr$, implying that xy is again a solution for the $(1, 2)$-permutational PCP. □

We stress that the undecidability of the (m, n)-permutational PCP was proved in [19], using the machinery of LBA's used already in [18], for all m and n. This extends the result is [18] where it was shown that the $(n, 1)$-permutational PCP is undecidable for all n[3].

The proof in [18] and [19] are rather involved because of the employment of the computations of LBA's, and there is a quest for simpler treatments of the problems. There is a line of new simplified proofs for Ruohonen's results

[2]Note that $(2, 2)$-permutational PCP is called the *PCP for circular words* in [18]

[3]Note that in [18] $(n.1)$-permutational PCP is called n-permutational PCP.

on the permutational PCP's, which use of a word problem of the special type of semi-Thue systems: For the case of $(2,1)$-permutational PCP a somewhat simpler proof was given in [9], where the problem was called the *circular PCP*. In [5] a simplified proof for $(n,1)$-permutational PCP (or n-permutational PCP) was given for all positive n. For $\mathbb{Z}$PCP a simpler proof was given in [12].

In the next section we give a new proof to undecidability of the conjugate-PCP by reducing it to the word problem for a special type of semi-Thue systems. These are is indeed the same special semi-Thue systems that were used for provingundecidability of the $\mathbb{Z}$PCP in [12], but the construction here is different due to differences in the $\mathbb{Z}$PCP and the conjugate-PCP. By Lemma 1, this also proves undecidability of the $(2,2)$-permutational PCP.

In the final section we show that the $\mathbb{Z}$PCP is in the class Σ_2^0 of the arithmetical hierarchy. This reflects to the result in [7], where it was proved that the infinite PCP is Π_1^0-complete as the PCP is known to be Σ_1^0-complete.

At first sight, it may seem that the $\mathbb{Z}$PCP and the conjugate PCP do not have anything in common, but that is not the case. In both of these problems, solutions have a shift, in the $\mathbb{Z}$PCP the shift makes the images, two bi-infinite words, equal and in the conjugate PCP the images are equal over a cyclic shift of the word. Therefore, the construction in the next section for the conjugate PCP has similar ideas as the construction for the $\mathbb{Z}$PCP in [12].

2 The proof of undecidability of the conjugate-PCP

We shall shortly recall the construction of the semi-Thue system $T_{\mathcal{M}}$ in [12]. First of all, a *semi-Thue system* T is a pair (Γ, R) where $\Gamma = \{a_1, a_2, \ldots, a_n\}$ is a finite alphabet, the elements of which are called *generators* of T, and the relation $R \subseteq \Gamma^* \times \Gamma$ is the set of *rules* of T. We write $u \to_T v$, if there exists a rule $(x,y) \in R$ such that $u = u_1 x u_2$ and $v = u_1 y u_2$ for some words $u_1, u_2 \in \Gamma^*$. We denote by $\to_T^*$ the reflexive and transitive closure of $\to_T$, and by $\to_T^+$ the transitive closure of $\to_T$. Note that the index T is omitted from the notation, when the semi-Thue system studied is clear from the context. If $u \to^* v$ in T, we say that there is a derivation from u to v in T.

In the *word problem* for semi-Thue systems, is it asked, for a given semi-Thue system T and words w and u, whether $w \to_T^* u$. In a *circular word problem* on the other hand, it is asked whether there exists a word u for a given semi-Thue

system T such that $u \to_T^+ u$.

In [12], a special kind of semi-Thue system was constructed which harnesses the structure of a given deterministic Turing machine. Assume that a *Turing machine*, TM for short, $\mathcal{M}$ is of the form $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F)$, where Q is a finite set of states, Σ is a finite input alphabet, Γ is a finite tape alphabet satisfying $\Sigma \subseteq \Gamma$, containing a special blank symbol $\Box \in \Gamma \setminus \Sigma$, q_0 is a unique *initial state*, δ is a *transition* mapping from $Q \times \Gamma$ to subsets of $Q \times \Gamma \times \{L, R, S\}$, and $F \subseteq Q$ is the set of *accepting states*. We assume that in a TM, the transition mapping is a partial function as the TM's are assumed to be deterministic.

For purposes of this section we also assume that $F = \{H\}$, that is, there exists a unique accepting state $H \in Q$, called the *halting* state. It can be assumed that a computation of a TM halts (i.e., no more transitions are applicable) if and only it arrives to state H. We denote a configuration of $\mathcal{M}$ by a word $uqv\Box$, if the contents of the non-blank part of the tape is uv, and $\mathcal{M}$ is reading the first symbol of $v\Box$ in state $q \in Q$.

A semi-Thue system $S_{\mathcal{M}} = (\Lambda, R_S)$ imitating the computation of a fixed deterministic TM $\mathcal{M}$ is constructed using the following ideas originally given in [13]: $\Lambda = Q \cup \Sigma \cup \Gamma \cup \{L, R\}$, where L and R are end markers. Indeed, the initial configuration $q_0 w$ of $\mathcal{M}$ corresponds to a word $Lq_0wR \in \Lambda^+$ and the rules of R_S are implied by the transition function δ so that, for example,

$$(aqb, acp) \in R_S \text{ if } \delta(q, b) = (p, c, R),$$

and similarly for the other types of transitions. Now it is straightforward to see that a TM $\mathcal{M}$ halts on input w in the configuration uHv for some words u and v if and only if $Lq_0wR \to_{S_{\mathcal{M}}}^* LuHvR$. Since the halting problem of TM's on empty tape is undecidable, we may assume in the above that $w = \Box$.

We obtain a simple proof for undecidability of the the word problem, see [13], by adding letter-by-letter cancellation rules such that $LuHvR \to^* LHR$ to the semi-Thue system $S_{\mathcal{M}}$. Furthermore, by adding a special rule

$$(LHR, Lq_0wR) \tag{1}$$

we have a semi-Thue system with undecidable circular word problem. The semi-Thue system constructed is Q-*deterministic* meaning that in all rules (u, v), both u and v contain exactly one symbol from set Q.

In order to prove that the conjugate-PCP is undecidable, we need to modify the above construction a bit. First of all, we take another copy of the alphabet

Λ, say $\overline{\Lambda} = \{\overline{a} \mid a \in \Lambda\}$ and add also overlined copies of all rules except the rule (1) to the system. The special rule (1) is replaced by two new rules,

$$(LHR, \overline{Lq_0\Box R}) \text{ and } (\overline{LHR}, Lq_0\Box R). \tag{2}$$

Now the circular derivation

$$Lq_0\Box R \to^*_{S_{\mathcal{M}}} LuHvR \to^* LHR \to Lq_0\Box R$$

is transformed into circular derivation

$$Lq_0\Box R \to^* LuHvR \to^* LHR \to \overline{Lq_0\Box R} \to^* \overline{LuHvR} \to^* \overline{LHR} \to Lq_0\Box R$$

in our new system.

Finally, we simplify the alphabet Λ (and $\overline{\Lambda}$). Indeed, we encode injectively the letters in $\Lambda \setminus (Q \cup \{L, R\})$ into $\{a, b\}^+$, and denote the new alphabets $A = \{a, b, L, R\}$ and $B = Q$. We have now constructed a semi-Thue system $T_{\mathcal{M}} = (\Sigma, \mathcal{R})$ with the following properties:

1. $\Sigma = A \cup \overline{A} \cup B \cup \overline{B}$ with pairwise disjoint alphabets $A, \overline{A}, B, \overline{B}$. Notably $A = \{a, b, L, R\}$ where L, R are markers for the left and right border of the word, respectively.

2. $T_{\mathcal{M}}$ is $(B \cup \overline{B})$-deterministic in the following way:

   (i) $\mathcal{R} \subseteq (A^*BA^* \times A^*BA^*) \cup (\overline{A^*BA^*} \times \overline{A^*BA^*}) \cup (A^*BA^* \times \overline{A^*BA^*}) \cup (\overline{A^*BA^*} \times A^*BA^*)$.

   (ii) If t_i is a rule in $\mathcal{R}$ where none of the symbols are overlined, then the corresponding overlined rule $\overline{t_i}$, where all symbols are overlined is also in $\mathcal{R}$, and vice versa.

   (iii) For all words $w \in (A \cup \overline{A})^*(B \cup \overline{B})(A \cup \overline{A})^*$, if there is a rule in $\mathcal{R}$ giving $w \to_T w'$ then the rule is unique.

   (iv) There is a single rule from $A^*BA^* \times \overline{A^*BA^*}$ and a single rule from $\overline{A^*BA^*} \times A^*BA^*$, moreover these rules are such that they re-write everything between the markers L and R, namely if there are rules giving $u \to_{T_{\mathcal{M}}} \overline{w_0}$ and $\overline{u} \to_{T_{\mathcal{M}}} w_0$ for a $u \in A^*BA^*$ then the rules are $(u, \overline{w_0})$ and $(\overline{u}, w_0)$, respectively. These rules are the rules in (2) coded into Σ.

3. $T_{\mathcal{M}}$ has an undecidable circular word problem. In particular it is undecidable whether T has a circular derivation $w_0 \to^*_{T_{\mathcal{M}}} w_0$ where $w_0 \in A^*BA^*$ is the word appearing in the rules of 2(iv). Note that w_0 and u in the case 2(iv) are fixed words from the construction of the semi-Thue system $T_{\mathcal{M}}$ for a particular Turing machine M, and $w_0 \neq u$.

The special $(B \cup \overline{B})$-determinism of $T_{\mathcal{M}}$ can be interpreted as derivations being in two different phases: the normal phase and the overlined phase. Transitioning between phases happens via the unique rules from 2(iv). It is straightforward to see that all derivations do not go through phase changes and that the phase is changed more than once if and only if T has a circular derivation. The system considered is now fixed from the context and we write the derivations omitting the index T simply as $\to$.

We now add a few additional rules to $T_{\mathcal{M}}$: we remove the unique rule $(u, \overline{w_0})$ and replace it with one extra step by introducing rules (u, s) and $(s, \overline{w_0})$ where s is a new symbol for the intermediate step. The corresponding overlined rules $(\overline{u}, \overline{s})$ and $(\overline{s}, w_0)$ are added also to replace the rule $(\overline{u}, w_0)$. These new rules are needed in identifying the border between words u and v, and adding them has no effect on the behaviour of $T_{\mathcal{M}}$.

By the case 3 of the properties of $T_{\mathcal{M}}$ we have the following lemma.

Lemma 2. *Assume that the semi-Thue system $T_{\mathcal{M}}$ is constructed as in the the above. Then $T_{\mathcal{M}}$ has an undecidable individual circular word problem for the word w_0.*

We now reduce the individual circular word problem of the system $T_{\mathcal{M}}$ to the conjugate-PCP.

Let $\mathcal{R} = \{t_0, t_1, \ldots, t_{h-1}, t_h\}$, where the rules are $t_i = (u_i, v_i)$. We denote by l_x and r_x the left and right desynchronizing morphisms defined by

$$l_x(a) = xa, \qquad r_x(a) = ax$$

for all words x. In the following we consider the elements of $\mathcal{R}$ as letters. Denote by A_j the alphabet A where letters are given subscripts $j = 1$ and 2, respectively. Define the morphisms $h, g : (A_1 \cup A_2 \cup \overline{A_1} \cup \overline{A_2} \cup \{\#, \overline{\#}, I\} \cup \mathcal{R})^* \to \{a, b, d, e, f, \#, \$, \pounds\}^*$ according to the following table:

	h	g	
I	$\$l_{d^2}(w_0\#)d$	$\pounds ee,$	
x_1	dxd	$xee,$	$x \in \{a, b\}$
x_2	ddx	$xee,$	$x \in \{a, b\}$
t_i	$d^{-1}l_{d^2}(v_i)$	$r_{e^2}(u_i),$	$t_i \notin \{t_{h-1}, t_h\}$
t_{h-1}	$dsff$	$r_{e^2}(u\#)$	
t_h	$f\$\pounds l_{e^2}(w_0\#)ee$	$sfff\pounds\$dd$	
$\#$	$dd\#d$	$\#ee$	
$\overline{x_1}$	xee	$xdd,$	$\overline{x} \in \{\overline{a}, \overline{b}\}$
$\overline{x_2}$	exe	$xdd,$	$\overline{x} \in \{\overline{a}, \overline{b}\}$
$\overline{t_i}$	$e^{-2}l_{e^2}(v_i)e$	$r_{d^2}(u_i),$	$\overline{t_i} \notin \{\overline{t_{h-1}}, \overline{t_h}\}$
$\overline{t_{h-1}}$	sf	$r_{d^2}(u\#)$	
$\overline{t_h}$	$ff\pounds$	$sfff\$$	
$\overline{\#}$	$e\#ee$	$\#dd$	

Here the re-writing rules are of the form $t_i = (u_i, v_i)$, for u_i, v_i. The following rules play important roles:
$t_{h-1} = (u, s)$, where u is the unique word such that $(u, \overline{w_0}) \in R$, and
$t_h = (s, \overline{w_0})$.

We begin by examining the forms of the images of h and g. The morphisms are modified from the ones in [12] with slight alterations made such that it is possible to have (finite) solutions to the instance of the conjugate-PCP with easily identifiable borders between the factors u and v using special symbols \$ and £. The symbols d, e and f function as desynchronizing symbols. The desynchronizing symbols d and e make sure that in the solution w the factors that will represent the configurations of the semi-Thue system $T_{\mathcal{M}}$ are of correct form, that is of the form where determinism is kept intact. This follows from the forms of h and g: under g all images are desynchronized by either e^2 (non-overlined letters) or d^2 (overlined letters). To have similarly desynchronized factors in the image under h we note that in the pre-image the words between two #-symbols (similarly for overlined symbols $\overline{\#}$) are of the form $\alpha t \beta$ where $\alpha \in \{a_1, b_1\}$, $\beta \in \{a_2, b_2\}$ and $t \in \mathcal{R}$ (with end markers L and R omitted from α and β). The symbol f is not really used in desynchronizing but makes sure that the change between phases is carried out correctly.

The following lemma is useful in our proof:

Lemma 3. *The words $h(w)$ and $g(w)$ are conjugates if and only if $h(w_1)$ and*

$g(w_2)$ are conjugates for all conjugates w_1 and w_2 of w.

Proof. If $h(w_1)$ and $g(w_2)$ are conjugates for all conjugates w_1 and w_2 of w then of course $h(w)$ and $g(w)$ are conjugates.

Assume then that $h(w)$ and $g(w)$ are conjugates and let w_1 and w_2 be conjugates of w. There are then suffixes x and y of w such that $w_1 = xwx^{-1}$ and $w_2 = ywy^{-1}$. Denote $wx^{-1} = w'$ and $wy^{-1} = w''$. Now $h(w_1) = h(xw') = h(x)h(w')$ is a conjugate of $h(w')h(x) = h(w'x) = h(w)$ and $g(w_2) = g(yw'') = g(y)g(w'')$ is a conjugate of $g(w'')g(y) = g(w''y) = g(w)$. By our assumption also $h(w_1)$ and $g(w_2)$ are conjugates. □

Next we will show that a circular derivation beginning from a fixed word w_0 exists in $T_{\mathcal{M}}$ if and only if there is a solution to the conjugate-PCP instance (h, g). We prove the claim in the following two lemmata.

Lemma 4. *If there is a circular derivation in $T_{\mathcal{M}}$ beginning from w_0, then there exists a non-empty word w such that $h(w) \sim_2 g(w)$.*

Proof. Assume that a circular derivation exists. The derivation is then of the form $w_0 = \alpha_1 u_1 \beta_1 \to \alpha_1 v_1 \beta_1 = \alpha_2 u_2 \beta_2 \to \cdots \to u \to s \to \overline{w_0} = \overline{\alpha_1 u_1 \beta_1} \to \cdots \overline{u} \to \overline{s} \to w_0$, where s and u as defined earlier for $T_{\mathcal{M}}$. This derivation can be coded into a word

$$w = Iw_1\#w_2\#w_3\#\cdots\#t_{h-1}t_h\overline{w_1\#w_2\#w_3\#\cdots\#t_{h-1}t_h},$$

where $w_i = \alpha_i t_i \beta_i$ for each i, where $t_i = (u_i, v_i)$ is the unique rewriting rule used in each derivation step. The rules t_{h-1} and t_h appear right before transition to overlined part of the derivation as they correspond to the final and intermediate steps before the transition. Let us consider the images of w under the morphisms h and g defined in the above:

$$h(w) = \$l_{d^2}(w_0\#\alpha_1 v_1 \beta_1\#\alpha_2 v_2 \beta_2\#\cdots\#s)fff\$\pounds l_{e^2}(w_0\#\alpha_1 v_1 \beta_1 \cdots\#s)fff\pounds)$$

and

$$g(w) = r_{e^2}(\pounds\alpha_1 u_1 \beta_1\#\alpha_2 u_2 \beta_2\#\cdots\#u\#)sfff\pounds r_{d^2}(\$\alpha_1 u_1 \beta_1 \cdots\#u\#)sfff\$).$$

These images are indeed very similar. They match at all positions that do not contain a desynchronizing symbol (d or e) or a special symbol (\$ or £). Thus, if we erase all of these non-matching symbols we would have equality (and of the form q^2 for a word q). Also the non-matching symbols are such that d is always

matched with e and \$ is always matched with £. It is clear that the factors in both $h(w)$ and $g(w)$ beginning and ending with the same special symbol are the same, that is, the factor of the form $\$\cdots\$$ and the factor of the form $£\cdots£$ both images are equal. It follows that $h(w) \sim_2 g(w)$, which proves our claim. □

Lemma 5. *If there exists a non-empty word w such that $h(w) \sim_2 g(w)$, then there is a circular derivation in $T_{\mathcal{M}}$ beginning from w_0.*

Proof. Firstly we show that the factor f^3 must appear in $h(w)$ and hence $t_{h-1}t_h$ or $\overline{t_{h-1}t_h}$ has to be a factor in w. Assume on the contrary: there is no factor f^3 in $h(w)$.

From the construction of g we know that also $h(w)$ must be desynchronized so that between the letters there is either a factor d^2 or e^2. Conjugation of $g(w)$ does not break this property except possibly in the beginning and the end of $h(w)$ ($h(w)$ could start and end in a single desynchronizing symbol).

Take now the first letter c of w. We can assume that it is a non-overlined letter as the considerations are similar for the overlined case. The letter c cannot be t_{h-1} as it would have to be followed by t_h: f^2 does not appear as a factor under g without f^3, and $t_{h-1}\overline{t_h}$ produces f^4, which is uncoverable by g. From the construction of h we see that the letters following c must also be non-overlined, otherwise the desynchronization would be broken. Thus the desynchronizing symbol is the same for all the following letters. But as we can see from the form of the morphisms h and g, we have a different desynchronizing symbols under g for c and its successors. It is clear that $h(g)$ must contain both d and e and so w must have both non-overlined and overlined letters. If there is a change in the desynchronizing symbol in $h(w)$ then it contradicts the form of the images under g. Hence we must have the factor $t_{h-1}t_h$ in w to make the transition without breaking the desynchronization.

The images of the factor $t_{h-1}t_h$ are

$$h(t_{h-1}t_h) = dsfff\$£l_{e^2}(w_0\#)ee$$

and

$$g(t_{h-1}t_h) = r_{e^2}(u\#)sfff\$£dd.$$

As we can see the desynchronizing symbols do not match. Hence we also must have the overlined copy of this factor in w, that is a factor $\overline{t_{h-1}t_h}I$, the images of which are (the letter I is a forced continuation to the overlined factor to account for the special symbols \$ and £):

$$h(\overline{t_{h-1}t_h}I) = sfff\pounds\$l_{d^2}(w_0\#)d$$

and

$$g(\overline{t_{h-1}t_h}I) = r_{d^2}(u\#)sfff\$\pounds ee.$$

One of either of these factors has one swap between the symbols d and e. From the above we concluded that we need an even number of these swaps as for every factor $t_{h-1}t_h$ we must also have the factor $\overline{t_{h-1}t_h}I$ and vice versa. It is possible that $h(w)$ ends in the letter f. In this case the swap happens "from the end to the beginning", i.e., the prefix of a factor doing the swap is at the end of w and the remaining suffix is at the beginning of w. The following proposition shows that we can in fact restrict ourselves to the case where the factors $t_{h-1}t_h$ and $\overline{t_{h-1}t_h}$ are intact, that is, the swap does not happen from the end to the beginning of $h(w)$ as a result of the conjugation between $h(w)$ and $g(w)$. At this point we make an observation.

Observation. *It may be assumed that the first and last symbols of $h(w)$ are \$ and £.*

Indeed, if $h(w)$ is not of the desired form then it has £\$ as a factor (by above the symbols from $\overline{t_{h-1}t_h}I$ are in w). Images of the letters under h do not have £\$ as a factor so there is a factorization $w = w_1w_2$ such that $h(w_1)$ ends in £ and $h(w_2)$ begins with \$ (w_1 ends in $\overline{t_h}$ and w_2 begins with I). By Lemma 3, $h(w)$ and $g(w)$ are conjugates if and only if $h(w_2w_1)$ and $g(w_2w_1)$ are, where now $h(w_2w_1)$ has \$ as the first symbol and £ as the last symbol.

Now by the observation we may assume that w begins with I and ends with $\overline{t_h}$. From this it also follows that if $h(w) = uv$ and $g(w) = vu$ the word u has \$ as the first and the last symbol and v has £ as the first and the last symbol. It follows that $w = I \cdots t_h \cdots \overline{t_h}$, where the border between u and v is in the image $h(t_h)$:

$$\begin{aligned} h(w) &= \underbrace{\$l_{d^2}(w_0\#)d \quad \cdots \quad f\$\pounds l_{e^2}(w_0\#)ee}_{u} \underbrace{\quad \cdots \quad ff\pounds}_{v} \\ g(w) &= \underbrace{\pounds ee \quad \cdots \quad sfff\pounds\$dd}_{v} \underbrace{\quad \cdots \quad sfff\$}_{u} \end{aligned}$$

Here the border between u and v need not be in the image of the same instance of t_h. Nevertheless we know by above that in the image under g the

word u begins with $\$l_{d^2}(w_0\#)d$. To get this image as a factor of $g(w)$ we must have $t_h\overline{\alpha_1 t_1 \beta_1 \#}$ in w, where $t_1 = (u_1, v_1)$ is the first rewriting rule used and $w_0 = \alpha_1 u_1 \beta_1$. Now

$$h(t_h\overline{\alpha_1 t_1 \beta_1 \#}) = f\$\pounds l_{e^2}(w_0\#\alpha_1 v_1 \beta_1\#)ee$$

which shows that

$$I\alpha_1 t_1 \beta_1 \#\alpha_2 t_2 \beta_2 \# \text{ occurs in } w \tag{3}$$

where by the $(B \cup \overline{B})$-determinism of T the rule $t_2 \in \mathcal{R}$ is the unique rule and $\alpha_1, \alpha_2 \in L\{a_1, b_1\}^* \cup \{\varepsilon\}$ and $\beta_1, \beta_2 \in \{a_2, b_2\}^* R \cup \{\varepsilon\}$ are unique words such that $g(\alpha_2 t_2 \beta_2) = r_{e^2}(\alpha_2 u_2 \beta_2) = r_{e^2}(\alpha_1 v_1 \beta_1)$. Again,

$$h(I\alpha_1 t_1 \beta_1 \#\alpha_2 t_2 \beta_2 \#) = \$l_{d^2}(w_0\#\alpha_1 v_1 \beta_1 \#\alpha_2 v_2 \beta_2 \#)d$$

which is also a factor of $g(w)$ and implies that

$$t_h\overline{\alpha_1 t_1 \beta_1 \#\alpha_2 t_2 \beta_2 \#\alpha_3 t_3 \beta_3 \#} \text{ occurs in } w \tag{4}$$

for a unique $t_3 \in \mathcal{R}$ and $\alpha_3 \in L\{a_1, b_1\}^* \cup \{\varepsilon\}$, $\beta_3 \in \{a_2, b_2\}^* R \cup \{\varepsilon\}$.

We can see that the words given by this procedure beginning with I or t_h (as in 3 and 4, respectively) contain derivations of the system $T_{\mathcal{M}}$ starting from w_0 where configurations are represented as words between #-symbols and consecutive configurations in these words are also consecutive in $T_{\mathcal{M}}$ (as is explained in the beginning of the proof), that is, we get from the former to the latter by a single derivation step.

From the finiteness of w it follows that long enough factors of w of the forms 3 and 4 represent cyclic computations: the configuration s is reached eventually and from there we have the rule (s, w_0) which starts a new cycle. We conclude that $T_{\mathcal{M}}$ must have a cyclic computation starting from configuration w_0.

□

Lemmas 2, 4 and 5 together yield our main theorem:

Theorem 1. *The conjugate-PCP is undecidable.*

This result does not generalize to more complex $(1, n))$-permutations using this same construction by say, adding more desynchronizing symbols and border markers for each element in the permutation. The generalization of the conjugate-PCP would be the $(1, n)$-permutational PCP, stated below:

Problem (Image Permutation Post Correspondence Problem)**.** *Given two morphisms $h, g : A^* \to B^*$, does there exist a word $w \in A^+$ and an n-permutation σ such that $h(w) = u_1 u_2 \cdots u_n$ and $g(w) = u_{\sigma(1)} u_{\sigma(2)} \cdots u_{\sigma(n)}$ for some words $u_1, \ldots, u_n \in B^*$?*

The reason that our construction does not work for the general $(1, n)$-case is that allowing more factors to be permuted can force solutions that do not describe TM computations. This is because of special cases for different values of n and σ, but also by the fact that the permutated factors may be single letters. In fact any solution w that produces Abelian equivalent words $h(w)$ and $g(w)$ also has a permutation that makes one of the words into the other. A "simple" proof using the techniques in this chapter is for now deemed unlikely, and some other approach may prove to be more fruitful. Note that the undecidability of the Image Permutation PCP follows already from proof of Ruohonen for (m, n)-permutational PCP in [19].

As a related result we note that the PCP for the instances where one of the morphisms is a permutation of the other are undecidable. Indeed, it was shown by Halava and Harju in [8] that the PCP is undecidable for instances $(h, h\pi)$, where $h : A^* \to B^*$ is a morphism and $\pi : A^* \to A^*$ is a permutation.

3 Complexity of $\mathbb{Z}$PCP

In this section, we will consider the $\mathbb{Z}$PCP defined in the introduction. As mentioned, undecidability of the $\mathbb{Z}$PCP was proved in [12] using similar techniques than in the previous section for the conjugate-PCP. We begin by reformulating the problem in more details:

Problem 2 ($\mathbb{Z}$PCP)**.** *Let A be a finite alphabet. Given a finite set of pairs of words over A, say $(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)$, does there exist a bi-infinite sequence $\ldots i_{-k} \ldots i_{-1}, i_0, i_1, \ldots, i_k, \ldots$ of the indices such that*

$$\cdots u_{i_{-k}} \cdots u_{i_{-1}} u_{i_0} u_{i_1} \cdots u_{i_k} \cdots = \cdots v_{i_{-k}} \cdots v_{i_{-1}} v_{i_0} v_{i_1} \cdots v_{i_k} \cdots ?$$

The equality of two bi-infinite words is an equivalence of the sequences of symbols modulo a finite shift $s \in \mathbb{Z}$ in the positions of the sequences.

An instance of the $\mathbb{Z}$PCP is given by a finite set of pairs of words over A (which can be coded by an integer via a recursive coding) and a solution to this instance is a bi-infinite sequence $(i_k)_{k \in \mathbb{Z}} \in \{1, 2, \ldots, n\}^{\mathbb{Z}}$.

We shall need in the sequel the notion of a Turing machine reading infinite words. We now recall these notions.

The *first infinite ordinal* is ω. An ω-*word* over an alphabet Σ is an ω-sequence $a_1a_2a_3\cdots$, where for all integers $i \geq 1, \quad a_i \in \Sigma$. The *set of* ω-*words* over the alphabet Σ is denoted by Σ^ω. An ω-*language* over an alphabet Σ is a subset of Σ^ω. For an ω-word $\sigma = a_1a_2a_3\cdots$, we denote the prefix $a_1\cdots a_n$ by $\sigma[n]$.

As in the previous section, assume that a Turing machine $\mathcal{M}$ is of the form $\mathcal{M} = (Q, \Sigma, \Gamma, \delta, q_0, F)$, where $F \subseteq Q$ is the set of accepting states. Turing machines reading of infinite words have considered in [2, 20]. A Turing machine $\mathcal{M}$ accepts a word $\sigma \in \Sigma^\omega$ with the 2-*acceptance condition* iff there is an infinite run of $\mathcal{M}$ on input σ visiting infinitely often states from F. The 2-acceptance condition is also now known as the Büchi acceptance condition. On the other hand, a Turing machine $\mathcal{M}$ accepts a word $\sigma \in \Sigma^\omega$ with $2'$-*acceptance condition* iff there is an infinite run of $\mathcal{M}$ on σ visiting only finitely often the accepting states in F. The $2'$-acceptance condition is also now known as the co-Büchi acceptance condition.

We require in this article that an accepting run should be infinite on the input $\sigma \in \Sigma^\omega$, as in [20], and not that it is complete (i.e. we do not require that all the cells of the right-infinite tape of the Turing machine are visited nor that all letters of σ are read), or even non-oscillating, as in [2]. We refer the interested reader to [6] and papers cited in [20, 6] for a comparison between these modes of acceptance of infinite words by Turing machines.

We assume the reader to be familiar with the arithmetical hierarchy on subsets of $\mathbb{N}$, as a general reference we give [17, 15]. We now recall the definition of the arithmetical hierarchy on subsets of Σ^ω for a finite alphabet Σ, see [20]. An ω-language $L \subseteq \Sigma^\omega$ belongs to the class Σ^0_n iff there exists a recursive relation $R_L \subseteq \mathbb{N}^{n-1} \times \Sigma^\star$ such that

$$L = \{\sigma \in \Sigma^\omega \mid \exists x_1 Q_2 x_2 \ldots Q_n x_n \quad (x_1, \ldots, x_{n-1}, \sigma[x_n + 1]) \in R_L\},$$

where Q_i for $i = 2, \ldots, n$ is one of the quantifiers $\forall$ or $\exists$ (not necessarily in an alternating order). An ω-language $L \subseteq \Sigma^\omega$ belongs to the class Π^0_n iff its complement $\Sigma^\omega - L$ belongs to the class Σ^0_n. The inclusion relations that hold between the classes Σ^0_n and Π^0_n are the same as for the corresponding classes of the Borel hierarchy. The classes Σ^0_n and Π^0_n are strictly included in the respective classes $\mathbf{\Sigma}^0_n$ and $\mathbf{\Pi}^0_n$ of the Borel hierarchy.

An important result is that the modes of acceptance of ω-languages by deterministic Turing machines are connected to the classes of the arithmetical

hierarchy. In particular, an ω-language is in the arithmetical class Π_2^0 (respectively, Σ_2^0) if and only if it is accepted by a *deterministic* Turing machine with 2-acceptance condition, i.e. Büchi acceptance condition (respectively, with $2'$-acceptance condition, i.e. co-Büchi acceptance condition), see Corollary 2.3 in [20].

We now state the main result of this section.

Theorem 2. *The bi-infinite PCP is in the class* $\Sigma_2^0 \setminus \Pi_1^0$.

In the above statement the arithmetical classes refer to classes of sets of integers. Indeed, this means that the set of instances of the bi-infinite PCP having a solution can be recursively coded by a set of integers in the class $\Sigma_2^0 \setminus \Pi_1^0$.

Proof. We firstly show that the $\mathbb{Z}$PCP is not in the class Π_1^0. This is actually a direct consequence of the proof of the undecidability of the $\mathbb{Z}$PCP in [12]. Indeed, the proof shows that there exists a reduction of the halting problem for Turing machines to the $\mathbb{Z}$PCP. On the other hand, it is well known that the halting problem for Turing machines is Σ_1^0-complete, hence the $\mathbb{Z}$PCP is Σ_1^0-hard and, in particular, it is not in the class Π_1^0.

Secondly, we prove that the $\mathbb{Z}$PCP is in the class Σ_2^0. Let us consider an instance of the $\mathbb{Z}$PCP given by a finite set of pairs of words over A, where A is a finite alphabet, say $Ins = \{(u_1, v_1), (u_2, v_2), \ldots, (u_n, v_n)\}$.

In the first step, we are going to associate to this instance a deterministic Turing machine with co-Büchi acceptance condition which accepts exactly the codes of the solutions to the instance Ins. For this we encode a bi-infinite sequence of integers in $\{1, 2, \ldots, n\}$

$$\ldots i_{-k} \ldots i_{-2} i_{-1} i_0 i_1 i_2 \ldots i_k \ldots \tag{5}$$

of a possible solution of the instance Ins for the $\mathbb{Z}$PCP into a pair of infinite sequences

$$i_0 i_1 i_2 \ldots i_k \ldots \quad \text{and} \quad i_0 i_{-1} i_{-2} \ldots i_{-k} \ldots .$$

Now both of these sequences are infinite words over the alphabet $\{1, 2, \ldots, n\}$, so that we can code the bi-infinite sequence in (5) into an ω-word over the finite alphabet $\{1, 2, \ldots, n\} \times \{1, 2, \ldots, n\}$ so that $(i_j)_{j \in \mathbb{Z}}$ corresponds to

$$(i_0, i_0)(i_1, i_{-1})(i_2, i_{-2}) \ldots (i_k, i_{-k}) \ldots . \tag{6}$$

Next we show that the set of (codes of) solutions of the instance Ins of the $\mathbb{Z}$PCP are accepted by a deterministic Turing machine $\mathcal{M}$ with co-Büchi

acceptance condition, (with set of final states F for the co-Büchi acceptance condition). We informally explain the behaviour of this Turing machine:

The TM $\mathcal{M}$ works with integers s denoting the shift in the images. Indeed, first the shift $s = 0$. For an input of the form (6), denote by $u = \ldots u_{i_{-k}} \ldots u_{i_{-2}} u_{i_{-1}} u_{i_0} u_{i_1} u_{i_2} \ldots u_{i_k} \ldots$ and $v = \ldots v_{i_{-k}} \ldots v_{i_{-2}} v_{i_{-1}} v_{i_0} v_{i_1} v_{i_2} \ldots v_{i_k} \ldots$ There is a (possibly infinite test) we call TEST:

TEST: For $m = 0, 1 \ldots$, check that $u(m) = v(m+s)$ and $u(-m) = v(-m+s)$.

If the TEST fails and for some m one of the equations does not hold (that is, $\mathcal{M}$ found an error and the sequence is not a solution with the shift s), then $\mathcal{M}$ enters in some state in F, and sets $s := -s$ if the TEST is done odd number of times and $s := |s| + 1$ it is done even number of times, and does the TEST for that new s.

It is rather obvious that if $\mathcal{M}$ visits states of F only finitely many times, the sequence (6) codes a solution of the instance Ins as for some shift s $\mathcal{M}$ found no error, that is, $u = v$ modulo some shift $s \in \mathbb{Z}$. Indeed, then the TM $\mathcal{M}$ accepts the coding (6) of the bi-infinite sequence of integers (5) with co-Büchi acceptance condition. Conversely, if the sequence (6) codes a solution to the instance Ins of $\mathbb{Z}$PCP, then (6) is accepted by the TM $\mathcal{M}$ with co-Büchi acceptance condition.

We do not go into the details of defining $\mathcal{M}$ but note that it is an easy exercise to construct such a deterministic Turing machine with co-Büchi acceptance condition from the instance Ins.

Now the set of infinite words accepted by such a deterministic Turing machine with co-Büchi acceptance condition is known to be an effective Σ^0_2 set. Moreover, Cenzer and Remmel proved in [1, Theorem 4.1.(iii)] that the non-emptiness problem for such effective Σ^0_2 sets is in the class Σ^0_2. Thus the problem to determine whether a given instance Ins of the $\mathbb{Z}$PCP has a solution is in the class Σ^0_2. $\square$

The next goal of this study of the complexity of the $\mathbb{Z}$PCP would be to determine its exact complexity. In particular, is it located at the second level of the arithmetical hierarchy? Is it Σ^0_2-complete? We leave these questions as an open problem for further study.

References

[1] D. A. Cenzer and J. B. Remmel. Index sets for omega-languages. *Mathematical Logic Quaterly*, 49(1):22–33, 2003.

[2] R. Cohen and A. Gold. ω-computations on Turing machines. *Theoretical Computer Science*, 6:1–23, 1978.

[3] J. Dong and Q. Liu. Undecidability of infinite Post correspondence problem for instances of size 8. *RAIRO Theor. Inform. Appl.*, 46(3):451–457, 2012.

[4] A. Ehrenfeucht, J. Karhumäki, and G. Rozenberg. The (generalized) Post correspondence problem with lists consisting of two words is decidable. *Theoret. Comput. Sci.*, 21(2):119–144, 1982.

[5] M. Ernvall, V. Halava, and T. Harju. On the n-permutation Post correspondence problem. *Theoretical Computer Science*, 601:15–20, oct 2015.

[6] O. Finkel. Ambiguity of ω-languages of Turing machines. *Logical Methods in Computer Science*, 10(3:12):1–18, 2014.

[7] O. Finkel. The exact complexity of the infinite Post correspondence problem. *Inform. Process. Lett.*, 115(6-8):609–611, 2015.

[8] V. Halava and T. Harju. Some new results on Post correspondence problem and its modifications. *Bull. EATCS*, 73:131–141, 2001.

[9] V. Halava and T. Harju. New proof for the undecidability of the circular PCP. *Acta Inform.*, 50(5-6):331–341, 2013.

[10] V. Halava, T. Harju, and M. Hirvensalo. Binary (generalized) Post correspondence problem. *Theoret. Comput. Sci.*, 276(1-2):183–204, 2002.

[11] V. Halava, T. Harju, and J. Karhumäki. Decidability of the binary infinite Post correspondence problem. *Discrete Appl. Math.*, 130(3):521–526, 2003.

[12] V. Halava, T. Harju, and E. Sahla. A new proof for undecidability of the bi-infinite Post correspondence problem. *Fund. Inform.*, 154(1-4):167–176, 2017.

[13] G. Huet and D. Lankford. On the uniform halting problem for term rewriting systems. *Rapport Laboria 283, INRIA*, 1978.

[14] T. Neary. Undecidability in binary tag systems and the Post correspondence problem for five pairs of words. In *32nd International Symposium on Theoretical Aspects of Computer Science*, volume 30 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 649–661. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2015.

[15] P. Odifreddi. *Classical Recursion Theory, Vol I*, volume 125 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1989.

[16] E. L. Post. A variant of a recursively unsolvable problem. *Bull. Amer. Math. Soc.*, 52:264–268, 1946.

[17] H. Rogers. *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, New York, 1967.

[18] K. Ruohonen. On some variants of Post's correspondence problem. *Acta Inform.*, 19(4):357–367, 1983.

[19] K. Ruohonen. *A Note on Permutational Variants of Post's Correspondence Problem*. Number 46 in Tampere University of Technology, Dept of Electr. Eng., Mathematics, Report. Tampere University of Technology, 1984.

[20] L. Staiger. ω-languages. In *Handbook of formal languages, Vol. 3*, pages 339–387. Springer, Berlin, 1997.