RAIRO-Theor. Inf. Appl. 57 (2023) 7
https://doi.org/10.1051/ita/2023008

RAIRO - Theoretical Informatics and Applications
www.rairo-ita.org

ON BI-INFINITE AND CONJUGATE POST CORRESPONDENCE PROBLEMS

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

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 conjugate PCP we give an undecidability proof by reducing it to the word problem for a special type of semi-Thue systems and for the bi-infinite PCP we give a simple argument that it is in the class Σ_2^0 of the arithmetical hierarchy.

Mathematics Subject Classification. 03D35, 03D40, 03D55, 68R01.

Received September 9, 2022. Accepted July 12, 2023.

1. Introduction

The original formulation of the *Post correspondence problem* (PCP) by Emil Post in [12] is the following:

Problem 1.1 (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 [12]. 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 setting. 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 the 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 [2, 7], and undecidable for

Keywords and phrases: Bi-infinite words, conjugate words, post correspondence problem, undecidability.

[1] Institut de Mathématiques de Jussieu – Paris Rive Gauche CNRS, Université Paris Cité, Sorbonne Université, Paris, France.
[2] Department of Mathematics and Statistics, University of Turku, Turku, Finland.

[*] Corresponding author: vehalava@utu.fi

$n = 5$, see [11]. 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 [8], and undecidable for 8 pairs, see [1]. It has been proved that the infinite PCP is not "more complex" than the PCP with respect to the arithmetical hierarchy, see [4], 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 pProblem* ($\mathbb{Z}$PCP), where it is asked whether or not there exits a bi-infinite sequence of the indices such that the words agree. The morphic version of the $\mathbb{Z}$PCP is the following:

Problem 1.2. 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 1.3. 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:

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

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 1.4. 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 [13] using linearly bounded automata (LBA)[1]. Indeed, the conjugate-PCP was not directly proved in [13], although it is claimed so in [14]. Let us consider the terminology of Ruohonen in [14] in a bit more details: Let u and v be words, and denote

[1] Note that in [13] the $\mathbb{Z}$PCP is called *doubly infinite PCP*.

$u \sim_m v$ 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 $v = u_{\tau(1)} \cdots u_{\tau(m)}$.

Problem 1.5 ((m,n)-permutation PCP)**.** Given morphisms $g, h\colon A^* \to B^*$ does there exist 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 [14], it is mentioned that (1,2)-permutational PCP was proved to be undecidable in [13], but the problem is not explicitly mentioned there. On the other hand, the (2,2)-permutational PCP is shown to be undecidable in [13][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 1.6. *For morphisms $g, h\colon A^* \to B^*$, 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 undecidability of the (m, n)-permutational PCP was proved in [14], using the machinery of LBA's used already in [13], for all m and n. This extends the result in [13] where it was shown that the $(n, 1)$-permutational PCP[3] is undecidable for all n.

The proofs in [13] and [14] are rather involved because of the employment of the computations of LBA's, and there is a quest for simpler treatment of the problems. There is a line of new simpler proofs for Ruohonen's results on the permutational PCP's, which uses the word problem of a special type of semi-Thue system: For the case of $(2, 1)$-permutational PCP a somewhat simpler proof was given in [6], where the problem was called the *circular PCP*. In [3] a simplified proof for the $(n, 1)$-permutational PCP (or n-permutational PCP) was given for all positive n. For the $\mathbb{Z}$PCP a simpler proof was given in [9].

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 indeed the same special semi-Thue systems that were used for proving undecidability of the $\mathbb{Z}$PCP in [9], but the construction here is different due to differences in the $\mathbb{Z}$PCP and the conjugate-PCP. By Lemma 1.6, 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 should be reflected with the result in [4], 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 [9].

[2] Note that $(2, 2)$-permutational PCP is called the *PCP for circular words* in [13].
[3] Note that in [13] $(n, 1)$-permutational PCP is called n-permutational PCP.

2. Semi-Thue system for the circular word problem

We shall shortly recall the construction of the semi-Thue system $T_{\mathcal{M}}$ in [9]. 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, it is 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 [9], a special kind of semi-Thue system was constructed which harnesses the structure of a given deterministic Turing machine. 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 [10]: $\Lambda = Q \cup \Sigma \cup \Gamma \cup \{L, R\}$, 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 word problem, see [10], 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

$$(LHR, Lq_0\Box R) \tag{2.1}$$

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.

3. Modifications to the semi-Thue system

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 (2.1) to the system. The special rule (2.1) is replaced by two new rules,

$$(LHR, \overline{Lq_0\Box R}) \text{ and } (\overline{LHR}, Lq_0\Box R). \tag{3.1}$$

Now the circular derivation

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

is transformed into circular derivation

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

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 $w \rightarrow_T w'$ 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 $u \rightarrow_{T_\mathcal{M}} \overline{w_0}$ and $\overline{u} \rightarrow_{T_\mathcal{M}} w_0$ 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 (3.1) coded into Σ.
3. $T_\mathcal{M}$ has an undecidable circular word problem. In particular it is undecidable whether T has a circular derivation $w_0 \rightarrow^*_{T_\mathcal{M}} w_0$ 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 $\rightarrow$.

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 $T_\mathcal{M}$.

4. Undecidability of the conjugate-PCP

We are now ready to prove our main result. By the case 3 of the properties of $T_\mathcal{M}$ we have the following lemma.

Lemma 4.1. *Assume that the semi-Thue system $T_\mathcal{M}$ is constructed as in 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. It is clear that for letters a and b and any word u

$$al_x(ub) = r_x(au)b. \tag{4.1}$$

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 [9] 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 $\pounds$. 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 4.2. *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 4.3. *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

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

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:

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

and, using the equalities in the derivation for w_0 and (4.1),

$$\begin{aligned} g(w) &= r_{e^2}(£ \alpha_1 u_1 \beta_1 \# \alpha_2 u_2 \beta_2 \# \cdots \# u \#) sfff £ r_{d^2}(\$ \alpha_1 u_1 \beta_1 \# \cdots \# u \#) sfff \$ \\ &= r_{e^2}(£ w_0 \# \alpha_1 v_1 \beta_1 \# \cdots \# u \#) sfff £ r_{d^2}(\$ w_0 \# \alpha_1 v_1 \beta_1 \# \cdots \# u \#) sfff \$ \\ &= £ l_{e^2}(w_0 \# \alpha_1 v_1 \beta_1 \# \cdots \# u \# s) fff £ \$ l_{d^2}(w_0 \# \alpha_1 v_1 \beta_1 \# \cdots \# u \# s) fff \$. \end{aligned}$$

Clearly, $h(w) \sim_2 g(w)$, which proves our claim. □

Lemma 4.4. *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

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

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

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

Either one 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 the 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 4.2, $h(w)$ 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 $h(t_h)$:

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

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

$$I\alpha_1 t_1 \beta_1 \# \alpha_2 t_2 \beta_2 \# \text{ occurs in } w \tag{4.2}$$

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

$$t_h \overline{\alpha_1 t_1 \beta_1 \# \alpha_2 t_2 \beta_2 \# \alpha_3 t_3 \beta_3 \#} \text{ occurs in } w \tag{4.3}$$

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 4.2 and 4.3, 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 4.2 and 4.3 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 4.1, 4.3 and 4.4 together yield our main theorem:

Theorem 4.5. *The conjugate-PCP is undecidable.*

This result does not generalize to more complex $(1, n)$)-permutations using the 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 [14].

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 [5] 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.

5. 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 [9] using similar techniques than in the previous section for the conjugate-PCP.

We will use a new modification of the $\mathbb{Z}$PCP called the *fixed shift* $\mathbb{Z}$PCP, and denoted by $\mathbb{Z}\mathrm{PCP}_S$, where S is a fixed integer.

Problem 5.1 ($\mathbb{Z}\mathrm{PCP}_S$)**.** Given two morphisms $h, g \colon A^* \to B^*$, does there exist a bi-infinite word w such that $h(w)(i) = g(w)(i + S)$ for all $i \in \mathbb{Z}$.

Clearly, $\mathbb{Z}\mathrm{PCP}_S$ is like the $\mathbb{Z}$PCP but in which a given instance $Ins = (h, g)$ has a solution for $\mathbb{Z}\mathrm{PCP}_S$ if and only if there is a bi-infinite word on which the two morphisms agree *via* the fixed shift S.

We may assume that the alphabet $A = \{1, 2, \ldots, n\}$ and consider a candidate solution w to the instance Ins:

$$w = \ldots i_{-k} \ldots i_{-2} i_{-1} i_0 i_1 i_2 \ldots i_k \ldots . \tag{5.1}$$

We divide w into two 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 (5.1) 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

$$(i_0, i_0)(i_1, i_{-1})(i_2, i_{-2}) \ldots (i_k, i_{-k}) \ldots . \tag{5.2}$$

Now these codes of the solutions of this instance for $\mathbb{Z}\mathrm{PCP}_S$ are the omega words accepted by a deterministic Turing machine (which can be constructed) with 1' acceptance condition similarly to the proof of complexity of infinite PCP in [4]. We omit the details here. It follows that the ω-language (*i.e.*, set of ω-words) accepted by such a machine is an effective Π^0_1 set. And the problem to determine whether such an effective Π^0_1 set is non-empty, is in the arithmetical class Π^0_1. Therefore, the fixed shift $\mathbb{Z}$PCP is in the class Π^0_1.

Now (h, g) has a solution for $\mathbb{Z}$PCP if and only if there exists S such that (h, g) has a solution for $\mathbb{Z}\mathrm{PCP}_S$. As existence of a solution for $\mathbb{Z}\mathrm{PCP}_S$ can be expressed by a Π^0_1 formula, it follows that the existence of a solution for $\mathbb{Z}$PCP can be expressed by Σ^0_2 formula.

On the other hand, the $\mathbb{Z}$PCP is not in the class Π^0_1. This is actually a direct consequence of the proof of the undecidability of the $\mathbb{Z}$PCP in [9]. Indeed, the proof shows that there exists a reduction of the halting problem for Turing machines to the $\mathbb{Z}$PCP. It is well known that the halting problem for Turing machines is Σ^0_1-complete, hence the $\mathbb{Z}$PCP is Σ^0_1-hard and, in particular, it is not in the class Π^0_1.

So we state the following theorem.

Theorem 5.2. *The bi-infinite PCP is in the class* $\Sigma^0_2 \setminus \Pi^0_1$.

Note that the above result can be achieved also defining fixed shift PCP for words of length $2n+1$ (as pointed by the referee). There the input is a pair of morphisms (h, g) and integers (s, n) and the test is whether for some word $w = i_{-n} \cdots i_{-1} i_0 i_1 \cdots i_n$, $i_j \in A$, $h(w)(k+s) = g(w)(k)$ for all defined k and $k+s$. This problem is clearly decidable for fixed (s, n) and the existence of a solution for the $\mathbb{Z}$PCP can now be expressed with $\exists s \forall n \psi(s, n)$ where ψ is the formula for the fixed shift PCP for words of length $2n+1$.

Finally, we conjecture that the complexity of the $\mathbb{Z}$PCP is in $\Sigma^0_2 \setminus \Pi^0_1 \cup \Sigma^0_1$, hence at the second level of the arithmetical hierarchy.

Acknowledgements. We would like to thank the anonymous referee for valuable comments that improved the quality of the article.

Funding information. Supported by emmy.network foundation under the aegis of the Foundation de Luxembourg.

References

[1] J. Dong and Q. Liu, Undecidability of infinite Post correspondence problem for instances of size 8. *RAIRO Theor. Inform. Appl.* **46** (012) 451–457.

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

[3] M. Ernvall, V. Halava and T. Harju, On the n-permutation Post correspondence problem. *Theoret. Comput. Sci.* **601** (2015) 15–20.

[4] O. Finkel, The exact complexity of the infinite Post correspondence problem. *Inform. Process. Lett.* **115** (2015) 609–611.
[5] V. Halava and T. Harju, Some new results on Post correspondence problem and its modifications. *Bull. EATCS* **73** (2001) 131–141.
[6] V. Halava and T. Harju, New proof for the undecidability of the circular PCP. *Acta Inform.* **50** (2013) 331–341.
[7] V. Halava, T. Harju and M. Hirvensalo, Binary (generalized) Post correspondence problem. *Theoret. Comput. Sci.* **276** (2002) 183–204.
[8] V. Halava, T. Harju and J. Karhumäki, Decidability of the binary infinite Post correspondence problem. *Discrete Appl. Math.* **130** (2003) 521–526.
[9] V. Halava, T. Harju and E. Sahla, A new proof for undecidability of the bi-infinite Post correspondence problem. *Fundam. Inform.* **154** (2017) 167–176.
[10] G. Huet and D. Lankford, On the uniform halting problem for term rewriting systems. Rapport Laboria 283, INRIA (1978).
[11] 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*. Vol. 30 of *LIPIcs. Leibniz Int. Proc. Inform.*. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern (2015) 649–661.
[12] E.L. Post, A variant of a recursively unsolvable problem. *Bull. Am. Math. Soc.* **52** (1946) 264–268.
[13] K. Ruohonen, On some variants of Post's correspondence problem. *Acta Inform.* **19** (1983) 357–367.
[14] K. Ruohonen, A Note on Permutational Variants of Post's Correspondence Problem, Vol. 46. Tampere University of Technology, Department of Electr. Eng., Mathematics, Report. Tampere University of Technology (1984).