Reduction Tree of the Binary Generalized Post Correspondence Problem

Vesa Halava and Štěpán Holub

Abstract

An instance of the (Generalized) Post Correspondence Problem is during the decision process typically reduced to one or more other instances, called its successors. In this paper we study the reduction tree of GPCP in the binary case. This entails in particular a detailed analysis of the structure of end blocks. We give an upper bound for the number of end blocks, and show that even if an instance has more than one successor, it can nevertheless be reduced to a single instance. This, in particular, implies that binary GPCP can be decided in polynomial time.

The Post Correspondence Problem, PCP for short, is one of the most important undecidable problems in the theory of computation. The PCP has been very useful in proving undecidability results in automata and formal language theory, in matrix theory etc. The PCP was proved undecidable by E. Post in 1946, see [14], in the following form: given two lists of words $(u_1, u_2, \ldots, u_n)$ and $(v_1, v_2, \ldots, v_n)$, the task is to determine whether or not there exists a nonempty sequence $i_1, i_2, \ldots, i_k$, such that

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

Such a sequence $i_1, i_2, \ldots, i_k$ is called a solution of the problem instance.

In the theory of computability it is interesting to study the borderline between the decidability and undecidability by relaxing and/or tightening requirements of a problem. It is known, taking PCP as an example, that if the lists of words have cardinality at most two, i.e. $n \leq 2$, then the problem is decidable (see [3] and [5]). On the other hand, the PCP is known to be undecidable for $n \geq 7$ (see [11]).

The binary PCP was originally proved decidable by Ehrenfeucht, Karhumäki and Rozenberg in [3] and by Pavlenko in [13]. However, the proof from [3] contains a gap (see p. 14 of the present paper for details); a full proof based on a similar approach is given by Halava, Harju and Hirvensalo in [5]. The basic idea is to transform a (non-periodic) instance into an instance of marked binary *Generalized PCP* (GPCP for short). In GPCP we are given two lists of words

$(u_1, u_2, \ldots, u_n)$ and $(v_1, v_2, \ldots, v_n)$ and words p_1, p_2, s_1, s_2 and the task is to determine whether or not there exists a nonempty sequence $i_1, i_2, \ldots, i_k$, such that

$$p_1 u_{i_1} u_{i_2} \cdots u_{i_k} s_1 = p_2 v_{i_1} v_{i_2} \cdots v_{i_k} s_2.$$

The instance of the PCP or the GPCP is called marked if the words u_1,$u_2, \ldots, u_n$ begin with n different letters, and similarly the words $v_1, v_2, \ldots, v_n$.

After the transformation into the marked GPCP, the decision is done by reducing the instance to simpler instances, called successors. The reduction can be iterated whence the successors form a directed tree. The main difficulty of the whole process are words s_1 and s_2 (called the *end block*), since their length may grow. In [5] it was proved that the length of the end blocks can be bounded, which yields a finite successor tree. Its size, however, is both in [3] and in [5] possibly exponential.

In this paper, we shall prove that it is possible to construct a successor tree that is nearly linear, that is, it does not contain any branching, except on the last level. This is achieved by compressing all possible successors into a unique instance. This indicates that the binary GPCP can be solved in polynomial time.

1 Preliminaries

1.1 Words

We shall first introduce some basic definitions and notations. As a general reference to combinatorics on words, we give Choffrut and Karhumäki [2].

An *alphabet* A is a finite nonempty set of symbols, which are usually called *letters*. A *word* over A is a finite sequence of symbols in A. We denote by A^* the monoid of all words over A endowed with the operation of concatenation. The *empty word*, denoted by ε, is the unit in A^*. The semigroup of all nonempty words is denoted by A^+. The *length* of the word u is denoted by $|u|$.

A word $u \in A^*$ is said to be a *prefix* of a word $v \in A^*$ if there exists $w \in A^*$ such that $v = uw$. This will be denoted by $u \leq v$. The prefix relation $\leq$ is the natural partial ordering on words. Therefore, if we speak about a minimal word, we refer to the prefix ordering. The first letter of a nonempty word v is denoted by $\mathrm{pref}_1(v)$. We say that u and v are *prefix-comparable*, or simply *comparable*, if $u \leq v$ or $v \leq u$, and denote it by $u \bowtie v$. The maximal common prefix of two words, u and v, is denoted by $u \wedge v$.

A word $u \in A^*$ is said to be a *suffix* of a word $v \in A^*$ if there exists $w \in A^*$ such that $v = wu$. If either u is a suffix of v or vice versa we say that u and v are *suffix-comparable*. The maximal common suffix of words u and v is denoted by $u \wedge_s v$.

A prefix (suffix resp.) u of v is called *proper* if $u \neq v$.

To simplify our exposition we shall sometimes consider the extension of the free monoid A^* to the free group generated by A. For example, if $u = vw$ then we can write $v = uw^{-1}$, $w = v^{-1}u$ but also $v^{-1} = wu^{-1}$. By $u \in (A^{-1})^*$ we

shall denote the fact that u is an element of the free monoid generated by letters a^{-1} and b^{-1}.

If $v = wuz$, then u is called a *factor* of v. Note that also prefixes and suffixes are factors.

We shall also work with the monoid $A^* \times A^*$ with the product defined by $(u, v)(u', v') = (uu', vv')$. Although this monoid is not free, the notions of prefix and suffix are well defined also here. For example the prefix ordering on pairs of words is given by $(u, v) \leq (u', v')$ if and only if $u \leq u'$ and $v \leq v'$.

Let $u = \ell_1, \ell_2, \ldots \ell_d$, where ℓ_i, $i = 1, 2, \ldots, d$, are letters. Then the *mirror image* of the word u, denoted by $\overline{u}$, is obtained by inverting the order of the letters, i.e.,

$$\overline{u} = \ell_d \ell_{d-1} \ldots \ell_1 .$$

Let A and B be two alphabets. A mapping g from A^* into B^*, that is, $g\colon A^* \to B^*$, is called a *morphism* if, for all $u, v \in A^*$,

$$g(uv) = g(u)g(v). \tag{1}$$

If $g(a) \neq \varepsilon$ for all $a \in A$, then g is said to be *nonerasing*. A morphism $g\colon A^* \to B^*$ is called *periodic* if there exists a word $w \in B^*$ such that $g(u)$ is a power of w for all $u \in A^*$.

A morphism $g : A^* \to B^*$ is *marked* if it is nonerasing and the images of the letters begin with different letters, i.e., if $g(a) \wedge g(b) = \varepsilon$ for all $a, b \in A$ such that $a \neq b$.

The *equality set* of morphisms g, h is the set

$$E(g, h) = \{h(w) = g(w) \mid w \in A^*\} .$$

In [8] and [9] it was proved that in the binary case the equality set of two non-periodic morphisms is either generated by just one word, or it is of the form $\{ba^i, a^i b\}$.

Let $g : A^* \to B^*$ be an arbitrary morphism. The *mirror image* of g is the morphism denoted by $\overline{g}$ and defined by

$$\overline{g}\,(x) = \overline{g(x)}\,,$$

for each $x \in A$. Note that in general $\overline{g}\,(u)$ does not equal to $g(\overline{u}\,)$ nor to $\overline{g(u)}$. Instead

$$\overline{g}\,(\overline{u}\,) = \overline{g(u)}\,.$$

We say that words $w_1, w_2, \ldots, w_n$ satisfy a *nontrivial relation* if an equality

$$w_{i_1} w_{i_2} \cdots w_{i_k} = w_{j_1} w_{j_2} \cdots w_{j_m}$$

holds, where all indeces are from $\{1, 2, \ldots, n\}$ and $(i_1, i_2, \ldots, i_k) \neq (j_1, j_2, \ldots, j_m)$.

We recall a basic fact from combinatorics on words.

Lemma 1. *If the words u and v satisfy a nontrivial relation, then they are powers of the same word.*

1.2 Binary Generalized PCP

The PCP is naturally and conveniently formulated using morphisms. Let $h, g : A^* \to B^*$ be morphisms, where $A = \{a_1, a_2, \ldots, a_n\}$ is an alphabet of n letters, and

$$g(a_i) = u_i \quad \text{and} \quad h(a_i) = v_i,$$

for $i = 1, 2, \ldots, n$. The PCP asks the question whether or not there exists a nonempty word $w \in A^+$ such that

$$g(w) = h(w). \tag{2}$$

The pair (g, h) is called an *instance* of the PCP, and a word w satisfying (2) is a *solution* of the instance. Note that the set of solutions of the instance $I = (g, h)$ is the equality set $E(I) = E(g, h)$, when including the empty word. Therefore, the PCP becomes defined also by asking to determine whether or not $E(I) = \{\varepsilon\}$.

Manipulation with PCP leads naturally to a modification of the problem called *Generalized Post Correspondence Problem*, GPCP for short. It consists of two morphisms $g, h\colon A^* \to B^*$ and four words $p_1, p_2, s_1, s_2 \in B^*$. The problem is to determine whether or not there exists a word $w \in A^*$ such that

$$p_1 g(w) s_1 = p_2 h(w) s_2.$$

The word w is again called a *solution* of the instance $I = ((p_1, p_2), g, h, (s_1, s_2))$ of the GPCP. Note that in the generalized PCP we admit the empty word as a solution, since it is not always true that $p_1 s_1 = p_2 s_2$.

The GPCP is undecidable in general, since instances of the PCP are also instances of the GPCP.

Denote by $\overline{I} = ((\overline{s_1}, \overline{s_2}), \overline{g}, \overline{h}, (\overline{p_1}, \overline{p_2}))$ *the mirror instance* of I. The next lemma is obvious.

Lemma 2. *A word w is a solution of I if and only if $\overline{w}$ is a solution of $\overline{I}$.*

In the *binary GPCP* the domain alphabet A is of size 2. We shall assume that $A = \{a, b\}$. It will be convenient to suppose that the image alphabet is also A. This assumption does not harm generality, since we can encode any alphabet injectively to the binary one.

Instances of GPCP split in two rather different types. If at least one of the morphisms is periodic we say that the whole instance is *periodic*. Otherwise the instance is *nonperiodic*. In this paper we shall deal exclusively with nonperiodic instances, since periodic ones are much easier to deal with. Accordingly, henceforth we shall suppose that $g, h\colon \{a, b\}^* \to \{a, b\}^*$ are nonperiodic morphisms. In particular, they are nonerasing.

By Lemma 1, if $uv = vu$, then the words u and v are powers of the same word. Therefore we have $g(ab) \neq g(ba)$, and we denote

$$z_g = g(ab) \wedge g(ba).$$

Similarly, we define z_h. Nonperiodic binary morphisms have the following property, which is fundamental for all our considerations. It can be found for example in [3]. Note that the following claims, formulated for g, hold analogously also for h.

Lemma 3. *Let au, bv be words such that $|g(au)| > |z_g|$, and $|g(bv)| > |z_g|$. Then*

$$g(au) \wedge g(bv) = z_g.$$

Proof. Let $z_g c$ be a prefix of $g(ab)$, and $z_g d$ a prefix of $g(ba)$ where c, d are distinct letters.

We show that $z_g c$ is a prefix of $g(a^i b)$ for each $i \geq 1$, and hence it is a prefix of $g(au)$. From $z_g c \leq g(aba)$ and $g(a)z_g \leq g(aba)$ we deduce $z_g c \leq g(a)z_g$. Let $i > 1$ and proceed by induction to obtain

$$z_g c \leq g(a)z_g \leq g(a)g(a^{i-1}b).$$

Similarly, we prove that $z_g d$ is a prefix of $g(bv)$, which completes the proof. □

The lemma says that the first letter of a word w is determined by any prefix of $g(w)$ strictly longer than $|z_g|$ (or, more precisely, by its $(|z_g|+1)$th letter). It also implies that for any two words u_1, u_2 we have

$$g(u_1)z_g \wedge g(u_2)z_g = g(u_1 \wedge u_2)z_g, \tag{3}$$

which easily yields that

$$u_1 \leq u_2 \quad \text{if and only if} \quad g(u_1)z_g \leq g(u_2)z_g. \tag{4}$$

We define the *marked version* $g_{\mathbf{m}}$ of g by

$$g_{\mathbf{m}}(a) = z_g^{-1} g(a) z_g, \qquad g_{\mathbf{m}}(b) = z_g^{-1} g(b) z_g.$$

Lemma 3 implies that $g_{\mathbf{m}}(a)$ is well defined, and

$$g_{\mathbf{m}}(w) = z_g^{-1} g(w) z_g$$

for all $w \in A^*$. Analogously, we define the marked version $h_{\mathbf{m}}$ of h.

Switching from a general morphism to a marked one is often convenient, and it constitutes the transposition from the binary PCP to the binary GPCP, mentioned in the introduction.

If z_g and z_h are suffix-comparable, then the word

$$\mathbf{c}(g,h) = z_h z_g^{-1} \tag{5}$$

is called the *critical overflow* of the morphisms g, h. Note that $\mathbf{c}(g,h) \in (A^{-1})^*$ if z_h is a suffix of z_g. By symmetry, however, we can suppose $\mathbf{c}(g,h) \in A^*$.

Note that if both morphisms are marked, then the critical overflow always exists, namely it is the empty word. The critical overflow has the following important property.

Lemma 4. *Let p and q be words, and let (u_1, v_1), (u_2, v_2) be pairs of words such that neither u_1 and u_2, nor v_1 and v_2 are comparable. If*

$$pg(u_i)z_g \bowtie qh(v_i)z_h$$

for both $i = 1, 2$, then

$$pg(u_1 \wedge u_2)z_g = qh(v_1 \wedge v_2)z_h.$$

Proof. Let w_i, $i = 1, 2$ denote the longer of the comparable words $pg(u_i)z_g$ and $qh(v_i)z_h$. The words u_1 and u_2 are not comparable, therefore neither $pg(u_1)z_g$ and $pg(u_2)z_g$ are, by (4). From $pg(u_1)z_g \leq w_1$ and $pg(u_2)z_g \leq w_2$ we deduce

$$w_1 \wedge w_2 = pg(u_1)z_g \wedge pg(u_2)z_g.$$

Similarly, we obtain

$$w_1 \wedge w_2 = qh(v_1)z_h \wedge qh(v_2)z_h.$$

Therefore,

$$pg(u_1)z_g \wedge pg(u_2)z_g = qh(v_1)z_h \wedge qh(v_2)z_h$$

and (3), used for both left and right side of the equality, completes the proof. □

Lemma 5. *Let morphisms g and h, and two words p and q be given. Then $A^* \times A^*$ contains at most one minimal pair of words (u, v) such that*

$$pg(u)z_g = qh(v)z_h. \tag{6}$$

If such words exist, the morphisms g and h have the critical overflow, and

$$pg(u) = qh(v)\mathbf{c}(g, h).$$

Proof. Recall that the minimality of (u, v) means that if $(u', v') \leq (u, v)$ and (u', v') satisfy (6), then $(u', v') = (u, v)$.

Suppose that

$$pg(u_i)z_g = qh(v_i)z_h$$

holds for two minimal pairs of words (u_i, v_i), $i = 1, 2$, such that $(u_1, v_1) \neq (u_2, v_2)$. First, we claim that u_1 and u_2 are incomparable. Suppose the contrary, and let, w.l.o.g., $u_1 \leq u_2$. Then Lemma 4 implies $pg(u_1)z_g \leq pg(u_2)z_g$, therefore also $qh(v_1)z_h \leq qh(v_2)z_h$, and $v_1 \leq v_2$; a contradiction with the minimality of (u_2, v_2). Analogously, we prove that v_1 and v_2 are incomparable.

Lemma 4 now yields

$$pg(u_1 \wedge u_2)z_g = qh(v_1 \wedge v_2)z_h,$$

a contradiction with the minimality of (u_1, v_1) and (u_2, v_2) again. The first part of the claim is proven.

The equality

$$pg(u) = qh(v)\mathbf{c}(g, h)$$

follows from the definition of the critical overflow. □

Let's explain how the previous lemma justifies the term "critical". Consider an instance $I = ((p_1, p_2), g, h, (s_1, s_2))$ of the GPCP. Suppose we are interested not in its solutions, but in pair of words (u, v) such that

$$p_1 g(u) s_1 = p_2 h(v) s_2. \tag{7}$$

Note that in this situation we have a solution of I if $u = v$, and, conversely, if w is a solution of I, then the pair (w, w) satisfies (7). Lemma 5 says that the words (u, v) are unique until the first occurrence of the critical overflow. Moreover, if some prefixes of u and v are given, then the continuation is again deterministic until the next occurrence of the critical overflow.

Next lemma says what can happen *between* two occurrences of the critical overflow.

Lemma 6. *For each $c \in A$ there is at most one minimal pair of words (e_c, f_c) such that $e_c f_c \neq \varepsilon$, $\mathrm{pref}_1(e_c) = c$ and*

$$\mathbf{c}(g, h) g(e_c) = h(f_c) \mathbf{c}(g, h). \tag{8}$$

Moreover, if both (e_a, f_a) and (e_b, f_b) exist, then $\mathrm{pref}_1(f_a) \neq \mathrm{pref}_1(f_b)$.

Proof. It is obvious from a length argument that both words e_c and f_c have to be nonempty. Put $p = \mathbf{c}(g, h) g(c)$, $q = \varepsilon$. Lemma 5 implies that there is at most one minimal pair of words (e_c, f_c) satisfying assumptions.

Analogous considerations for $p = \mathbf{c}(g, h)$, $q = h(c)$, $c \in A$, imply that $\mathrm{pref}_1(f_c) \neq \mathrm{pref}_1(f_c)$. □

The pairs (e_a, f_a) and (e_b, f_b) described in Lemma 6 will be called *letter blocks* of the pair of morphisms (g, h).

2 Successors

We are now approaching the central concept of the paper, which is based on Lemma 6. If both letter blocks (e_a, f_a) and (e_b, f_b) exist, we say that morphisms g and h *have a successor*, and define the *successor morphisms* $g_1, h_1 : A^* \to A^*$ of (g, h) by

$$\begin{aligned} g_1(a) &= e_a, & g_1(b) &= e_b, \\ h_1(a) &= f_a, & h_1(b) &= f_b. \end{aligned} \tag{9}$$

Lemma 6 immediately implies the following fact.

Lemma 7. *Successor morphisms are marked.*

From now on, if we speak about I, we suppose that I is an instance of the binary GPCP, and $I = ((p_1, p_2), g, h, (s_1, s_2))$, thereby fixing the meaning of variables p_1, p_2, s_1 and s_2.

We say that I *has a successor*, if morphisms g and h have a successor and, moreover, there are words p_1', p_2', s_1' and $s_2' \in A^*$ such that $p_1' \bowtie p_2'$, $\overline{s_1'} \bowtie \overline{s_2'}$, and

$$\begin{aligned} p_1 g(p_1') &= p_2 h(p_2') \mathbf{c}(g,h), \\ \mathbf{c}(g,h) g(s_1') s_1 &= h(s_2') s_2. \end{aligned} \tag{10}$$

The corresponding *successor instance* of I is then defined by

$$I' = ((p_1', p_2'), g_1, h_1, (s_1', s_2')).$$

The pair (p_1', p_2') is called a *beginning block*, if it is minimal. Similarly, if $(\overline{s_1'}, \overline{s_2'})$ is minimal, then the pair (s_1', s_2') is called an *end block*. Note that the beginning block is unique, by Lemma 5. However, in general, there can be several end blocks.

In order to explain the meaning of the successors let us develop the exposition preceding Lemma 6. Suppose that a pair (u, v) satisfies (7). Lemma 5 tells us that the words u and v are unique, except for the situations creating the critical overflow, that is, except for the prefixes (u', v') of (u, v) satisfying $p_1 g(u') = p_2 h(v') \mathbf{c}(g, h)$. Moreover, Lemma 6 guarantees that between two successive critical overflows we can expect letter blocks. Consequently, any pair (u, v) satisfying (7) that induces at least one critical overflow can be decomposed into beginning block, end block, and letter blocks. We therefore have the following definition.

Let w be a solution of I. We say that w is *decomposable* if there is a sequence

$$(u_0, v_0), (u_1, v_1), \ldots, (u_k, v_k), \tag{11}$$

with $k \geq 1$, such that

$$w = u_0 u_1 \cdots u_k = v_0 v_1 \cdots v_k,$$

and (u_0, v_0) is the beginning block of I, (u_i, v_i) with $i = 1, 2, \ldots, k-1$ are letter blocks of (g, h), and (u_k, v_k) is an end block of I.

The sequence (11) is called a *block decomposition* of w.

Lemma 8. *Let w be a solution of an instance I. If there are prefixes w_1 and w_2 of w such that*

$$p_1 g(w_1) = p_2 h(w_2) \mathbf{c}(g, h),$$

then w is decomposable, and its block decomposition is unique.

Proof. Lemma 5 implies that there is a unique beginning block. By Lemma 6, the letter blocks are minimal solutions to $\mathbf{c}(g,h) g(u_i) = h(v_i) \mathbf{c}(g,h)$, therefore the structure of the letter blocks in the decomposition is unique. The proof is completed by the requirement, that the end block is suffix-minimal. □

The above definition of successor morphisms and successor instance is motivated precisely by the block decomposition of decomposable solutions. If the

beginning block of I is denoted by (p'_1, p'_2) and one of its end blocks by (s'_1, s'_2), then it remains to determine the sequence of letter blocks between the beginning and the end block. This essentially means that we lift the considerations from the values of morphisms g and h to its arguments. It turns out that we obtain another instance of GPCP, namely the successor instance. The idea is formulated in the following lemma.

Lemma 9. *Assume that morphisms g, h have successor morphisms g_1 and h_1.*

If w is a decomposable solution of I, then there is a successor instance $I' = ((p'_1, p'_2), g_1, h_1, (s'_1, s'_2))$ of I such that

$$w = p'_1 g_1(w') s'_1 = p'_2 h_1(w') s'_2,$$

where w' is a solution of I'.

Conversely, if a successor instance $I' = ((p'_1, p'_2), g_1, h_1, (s'_1, s'_2))$ of I has a solution w', then

$$w = p'_1 g_1(w') s'_1 = p'_2 h_1(w') s'_2$$

is a decomposable solution of I.

Proof. The proof is a review of the previous definitions.

Let w be a decomposable solution of I, and let $(u_0, v_0), (u_1, v_1), \ldots, (u_k, v_k)$ be its (unique) block decomposition. Then we define a successor instance I' by $((u_0, v_0), g_1, h_1, (u_k, v_k))$. Let (e_a, f_a) and (e_b, f_b) be letter blocks of g and h, and let g_1 and h_1 be defined by (9). The word w' is now defined by $w' = c_1 c_2 \cdots c_{k-1}$, where

$$c_i = \begin{cases} a, & \text{if } (u_i, v_i) = (e_a, f_a)\,, \\ b, & \text{if } (u_i, v_i) = (e_b, f_b). \end{cases}$$

Clearly,

$$w = u_0 g_1(w') u_k = v_0 h_1(w') v_k,$$

which completes the first part of the proof.

Conversely, let $w' = \ell_1 \ell_2 \cdots \ell_d$ be a solution of $I' = ((p'_1, p'_2), g_1, h_1, (s'_1, s'_2))$, where ℓ_i are letters. The definition of morphisms g_1 and h_1 implies that

$$\mathbf{c}(g,h)\, g \circ g_1(w') = h \circ h_1(w')\, \mathbf{c}(g,h),$$

and (10) yields

$$p_1 g(p'_1 g_1(w') s'_1) s_1 = p_2 g(p'_2 g_1(w') s'_2) s_2.$$

Therefore w is a solution of I, and

$$(p'_1, p'_2), (g_1(\ell_1), h_1(\ell_1)), (g_1(\ell_2), h_1(\ell_2)), \ldots, (g_1(\ell_d), h_1(\ell_d)), (s'_1, s'_2)$$

is its block decomposition. □

The following lemma is dedicated to properties of mirror images of morphisms g and h, and their successors. The mirror images will play an important role in the study of end blocks.

Lemma 10. *Let g_1 and h_1 be successor morphisms of morphisms g and h. Then the morphisms $\overline{g}$ and $\overline{h}$ also have successors g'_1 and h'_1 such that (up to renaming of letters)*

$$g'_1(a) = \overline{re_ar^{-1}}, \qquad g'_1(b) = \overline{re_br^{-1}},$$
$$h'_1(a) = \overline{qf_aq^{-1}}, \qquad h'_1(b) = \overline{qf_bq^{-1}},$$

where r (q resp.) is the maximal common suffix of e_ae_b and e_be_a (f_af_b and f_bf_a resp.). Moreover,

$$g(r) = \overline{\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,)}\,h(q)\mathbf{c}(g,h). \tag{12}$$

Proof. Since $\mathrm{pref}_1(e_a) \neq \mathrm{pref}_1(e_b)$, the words e_a and e_b do not commute by Lemma 1. Therefore r is a proper suffix of both e_ae_b and e_be_a. Similarly for q.

Note that $\overline{r} = z_{\overline{g_1}}$, $\overline{q} = z_{\overline{h_1}}$, and

$$g'_1(c) = z^{-1}_{\overline{g_1}}\,\overline{g_1}\,(c)z_{\overline{g_1}}, \qquad h'_1(c) = z^{-1}_{\overline{h_1}}\,\overline{h_1}\,(c)z_{\overline{h_1}}$$

for $c \in \{a,b\}$. Therefore, g'_1 is the marked version of $\overline{g_1}$, and h'_1 the marked version of $\overline{h_1}$, which certifies that the morphisms g'_1 and h'_1 are well defined.

We have to show that g'_1 and h'_1 are the successors of $\overline{g}$ and $\overline{h}$, that is, we prove that

$$\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,)\,\overline{g}\,(\,\overline{r^{-1}}\;\overline{e_a}\;\overline{r}\,) = \overline{h}\,(\,\overline{q^{-1}}\;\overline{f_a}\;\overline{q}\,)\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,), \tag{13}$$
$$\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,)\,\overline{g}\,(\,\overline{r^{-1}}\;\overline{e_b}\;\overline{r}\,) = \overline{h}\,(\,\overline{q^{-1}}\;\overline{f_b}\;\overline{q}\,)\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,), \tag{14}$$

and that the pairs of words $(\,\overline{r^{-1}}\;\overline{e_a}\;\overline{r}\,,\,\overline{q^{-1}}\;\overline{f_a}\;\overline{q}\,)$ and $(\,\overline{r^{-1}}\;\overline{e_b}\;\overline{r}\,,\,\overline{q^{-1}}\;\overline{f_b}\;\overline{q}\,)$ are minimal satisfying (13) and (14) respectively.

Consider morphisms $\overline{g}$ and $\overline{h}$. For any $i \in \mathbb{N}$ we have

$$\overline{g}\,(\,\overline{e_a}\;\overline{e_b}\,)^i\,\overline{\mathbf{c}(g,h)} = \overline{\mathbf{c}(g,h)}\;\overline{h}\,(\,\overline{f_a}\;\overline{f_b}\,)^i,$$
$$\overline{g}\,(\,\overline{e_b}\;\overline{e_a}\,)^i\,\overline{\mathbf{c}(g,h)} = \overline{\mathbf{c}(g,h)}\;\overline{h}\,(\,\overline{f_b}\;\overline{f_a}\,)^i.$$

Lemma 4 and the definition of r and q imply that the critical overflow $\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,) = z_{\overline{h}}\,z^{-1}_{\overline{g}}$ exists, and

$$\overline{g}\,(\,\overline{r}\,) = \overline{\mathbf{c}(g,h)}\;\overline{h}\,(\,\overline{q}\,)\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,). \tag{15}$$

Using (15) and the equality $\mathbf{c}(g,h)g(e_a) = h(f_a)\mathbf{c}(g,h)$ we obtain

$$\overline{g}\,(\,\overline{e_a}\;\overline{r}\,) = \overline{g}\,(\,\overline{e_a}\,)\,\overline{\mathbf{c}(g,h)}\;\overline{h}\,(\,\overline{q}\,)\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,) = \overline{\mathbf{c}(g,h)}\;\overline{h}\,(\,\overline{f_a}\;\overline{q}\,)\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,),$$

and, applying (15) again,

$$\overline{g}\,(\,\overline{r^{-1}}\;\overline{e_a}\;\overline{r}\,) = \mathbf{c}(\,\overline{g}\,,\,\overline{h}\,)^{-1}\,\overline{h}\,(\,\overline{q^{-1}}\;\overline{f_a}\;\overline{q}\,)\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,).$$

Therefore

$$\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,)g(\,\overline{r^{-1}}\;\overline{e_a}\;\overline{r}\,) = h(\,\overline{q^{-1}}\;\overline{f_a}\;\overline{q}\,)\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,).$$

This completes the proof of (13). The equality (14) is obtained analogously.

Thus the pairs of $(\,\overline{r^{-1}}\;\overline{e_a}\;\overline{r}\,,\,\overline{q^{-1}}\;\overline{f_a}\;\overline{q}\,)$ and $(\,\overline{r^{-1}}\;\overline{e_b}\;\overline{r}\,,\,\overline{q^{-1}}\;\overline{f_b}\;\overline{q}\,)$ are products of letter blocks of $(\,\overline{g}\,,\,\overline{h}\,)$. It remains to show that they are minimal. Suppose that they are not. Then the pair $(\,\overline{g}\,,\,\overline{h}\,)$ has shorter letter blocks than (g,h); more precisely, the sum of lengths of the four letter block words of $(\,\overline{g}\,,\,\overline{h}\,)$ is strictly smaller than in case of (g,h). But (g,h) is the mirror image of $(\,\overline{g}\,,\,\overline{h}\,)$, and we can use the first part of the proof to obtain shorter letter blocks for (g,h), a contradiction.

Since the equality (12) is a mirror image of (15), we are done. □

3 Reduction tree

In this section we prove main results of the paper. As already noted, the basic difficulty in deciding the GPCP is the existence of letter blocks and successors; the decision is then reduced to the set of successors. Of course, one cannot exclude that a successor instance has again successors. This way we obtain a directed tree of successors and the decision procedure depends essentially on the shape of the tree. This encompasses two different questions: the depth and the width of the tree. The depth problem, or, viewed differently, the problem of the height of the tower of successors, is left aside in this paper. It can be faced by studying a property called "suffix complexity", which was introduced already in [3]. Detailed study of the suffix complexity in the binary case can be found in [10].

In this paper we deal with the width of the reduction tree, that is, with the number of successors an instance can have. So far in literature, it was admitted that there can be many successors, which implies that the reduction tree can be very wide, and the number of the instances to be decided threatens to grow exponentially. While, as we have seen, letter blocks and the beginning block are unique, the source of problems is the end block, which is not given uniquely.

In the present section we describe the structure of end blocks, and show that all successors of an instance, if there are more than one of them, can be replaced by just one "common" successor.

Recall that an end block of an instance I is a suffix-minimal pair of words (u,v), which satisfies

$$\mathbf{c}(g,h)g(u)s_1 = h(v)s_2,$$

and, moreover, (u,v) are suffix-comparable. For the successor instance only the value of vu^{-1} is relevant, which is either an element of A^* or $(A^{-1})^*$ according to whether u is a suffix of v or vice versa. In the first case, the pair (s_1', s_2') of the successor is equal to (ε, vu^{-1}), in the latter, we have $(s_1', s_2') = ((vu^{-1})^{-1}, \varepsilon) = (uv^{-1}, \varepsilon)$.

Therefore, we denote

$$\mathbf{e}(u,v) = \begin{cases} (\varepsilon, vu^{-1}), & \text{if } vu^{-1} \in A^*; \\ (uv^{-1}, \varepsilon), & \text{if } vu^{-1} \in (A^{-1})^+. \end{cases}$$

Note that

$$\mathbf{e}(uz, vz) = \mathbf{e}(uz', vz')$$

for any z, z', which is a natural way how to say that the two endblocks (uz, vz) and (uz', vz') are essentially the same.

Henceforth, by (e_a, f_a) and (e_b, f_b) we shall always denote letter blocks of (g, h), by r the maximal common suffix of $e_a e_b$ and $e_b e_a$, and by q the maximal common suffix of $f_a f_b$ and $f_b f_a$ (as in Lemma 10). The words r and q have the following property.

Lemma 11. *Either r is a proper suffix of both e_a and e_b, or q is a proper suffix of both f_a and f_b.*

Proof. Suppose the opposite. Then e_a and e_b are suffix-comparable, as well as f_a and f_b. Since $|g(e_a)| = |h(f_a)|$ and $|g(e_b)| = |h(f_b)|$, a length argument implies that either (e_a, f_a) is a suffix of (e_b, f_b), or vice versa. This contradicts minimality of letter blocks. □

Theorem 12. *Let both I and $\overline{I}$ have a successor. Then I has at most*

$$\max\{|e_a e_b|, |f_a f_b|\} + 1$$

end blocks. Furthermore, if (u, v) is an end block of I, then

$$ru = \overline{g_1'(z)}\, b_1, \qquad\qquad qv = \overline{h_1'(z)}\, b_2 \tag{16}$$

where $(\overline{b_1}, \overline{b_2})$ is the beginning block of $\overline{I}$, and z is a minimal word satisfying

$$r \leq \overline{g_1'(z)}\, b_1, \qquad\qquad q \leq \overline{h_1'(z)}\, b_2. \tag{17}$$

Proof. Let (u, v) be an end block. Then, by (12),

$$g(ru)s_1 = \overline{\mathbf{c}(\overline{g}, \overline{h})}\, h(qv)s_2$$

and therefore

$$ru = \overline{g_1'(z)}\, b_1, \qquad\qquad qv = \overline{h_1'(z)}\, b_2,$$

for some z. Using Lemma 10, this can be also written as

$$u = r^{-1}\, \overline{g_1'(z)}\, b_1 = g_1(\overline{z})r^{-1}b_1, \qquad v = q^{-1}\, \overline{h_1'(z)}\, b_2 = h_1(\overline{z})q^{-1}b_2.$$

Suffix minimality of the end block (u, v) implies that z is a minimal word satisfying (17).

If r is a suffix of both e_a and e_b, and q is a suffix of both f_a and f_b, then z is either empty or $z \in \{a, b\}$. Suppose that this is not the case. Let, for example, e_a and e_b be suffix-comparable. We let k denote the smallest integer such that r is a suffix of e_a^k and m the smallest integer such that r is a suffix of $(e_b)^m$. Note that z satisfies (17) as soon as $g_1'(z)$ is of length at least $e_a e_b$. Therefore either $m \leq 2$ (if $|e_a| \leq |e_b|$) or $k \leq 2$ (if $|e_b| \leq |e_a|$). It is not difficult to see that

$$z \in \{a^i b \mid i = 1, 2, \ldots k-1\} \cup \{b^j a \mid j = 1, 2, \ldots m-1\} \cup \{a^k, b^m\}. \qquad (18)$$

Since, clearly, we have $k, m \leq |r|$, the number of end blocks is limited by $|r| + 2$. Similarly, we obtain the bound $|q| + 2$, if the words f_a and f_b are suffix-comparable. This yields the desired bound $\max\{|e_a e_b|, |f_a f_b|\} + 1$. □

The following example illustrates the possibility of multiple end blocks.

Example 1. *Consider an instance with marked morphisms*

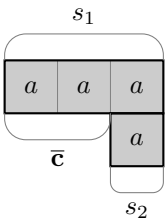
$$g(a) = a \qquad g(b) = b$$
$$h(a) = a \qquad h(b) = baa,$$

and $(s_1, s_2) = (aaa, a)$.

Then $(e_a, f_a) = (a, a)$, $(e_b, f_b) = (baa, b)$, $\mathbf{c}(g, h) = \varepsilon$, $\mathbf{c}(\overline{g}, \overline{h}) = aa$, $r = aa$ *and* $q = \varepsilon$. *Therefore*

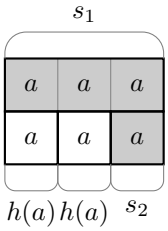
$$g_1(a) = a, \qquad g_1(b) = baa,$$
$$h_1(a) = a, \qquad h_1(b) = b.$$

The beginning block $(\overline{b}_1, \overline{b}_2)$ *of* $\overline{I}$ *is empty, since* $(\overline{s}_1, \overline{s}_2)$ *has already the critical overflow.*

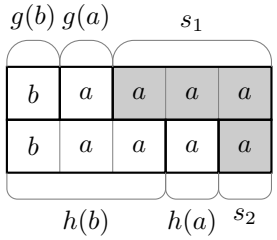


Consider the following three end blocks:

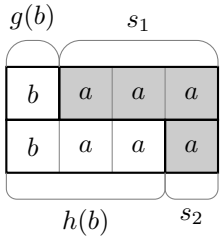
$$(u_1, v_1) = (\varepsilon, aa) = (g_1(aa)r^{-1}, h_1(aa)q^{-1})$$



$$(u_2, v_2) = (ba, ba) = (g_1(ba)r^{-1}, h_1(ba)q^{-1})$$



$$(u_3, v_3) = (b, b) = (g_1(b)r^{-1}, h_1(b)q^{-1})$$



The previous example should be compared with the claim in [3], p. 137, that there may be one or two successors. This claim is explained by a different definition of the "end block". The authors require that the end block is prefix-minimal, instead of suffix-minimal. Such a definition, however, does not allow to suppose that each solution has a block decomposition. Taking the previous example, a solution ending by the end block (ba, ba) does not have such a decomposition, since (ba, ba) is not a prefix-minimal end block. The claim, as well as the proof of Theorem 8.1 in [3], is therefore not correct.

Note, on the other hand, that in the previous example we have $\mathbf{e}(u_2, v_2) = \mathbf{e}(u_3, v_3) = (\varepsilon, \varepsilon)$. In fact, we conjecture that, although there can be more than two end blocks, there are at most two values of $\mathbf{e}(u, v)$.

The assumption of Theorem 12 regarding the mirror instance I is not superfluous as the following example shows.

Example 2. *Consider the instance* $I = ((a, \varepsilon), g, h, (\varepsilon, a))$ *where*

$$g(a) = aaa, \qquad g(b) = bab,$$
$$h(a) = abab, \qquad h(b) = aa.$$

We have $\mathbf{c}(g, h) = a$*, and there exist letter blocks* (aa, bbb) *and* (ba, ab)*. Moreover, both the beginning block and the end block exist; they are empty.*

However, the mirror instance $\overline{I}$ *has no beginning block. Indeed,* $\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,) = \varepsilon$ *and there are no suffix-comparable words* b_1*,* b_2 *satisfying*

$$g(b_1) = h(b_2)a.$$

Let us further investigate the situation when there are more successors.

Lemma 13. *Let both I and $\overline{I}$ have a successor, and let (u_1, v_1) and (u_2, v_2) be two distinct end blocks of I such that ru_i and qv_i are suffix-comparable, $i = 1, 2$. Then there is a word τ such that*

$$g(\tau)s_1 = \overline{\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,)}\,h(\tau)s_2, \tag{19}$$

and (τ, τ) is a suffix of each pair (ru, qv) where (u, v) is an end block of I and ru and qv are suffix-comparable.

Proof. Let (u_i, v_i), $i = 1, 2, \ldots k$ be all end blocks of I such that (ru_i, qv_i) are suffix-comparable. Let (g'_1, h'_1) be successor morphisms of $(\,\overline{g}\,,\,\overline{h}\,)$. Theorem 12 implies that

$$ru_i = \overline{g'_1(z_i)}\,b_1, \qquad\qquad qv_i = \overline{h'_1(z_i)}\,b_2$$

where $(\,\overline{b_1}\,,\,\overline{b_2}\,)$ is the beginning block of $\overline{I}$ and z_i, $i = 1, 2, \ldots k$, are prefix minimal. Let z be the longest suffix common to all z_i and suppose, w.l.o.g., that $z = z_1 \wedge z_2$. The words z_1 and z_2 are not comparable, therefore $z_1 = zz'_1$, $z_2 = zz'_2$ with $\mathrm{pref}_1(z'_1) \neq \mathrm{pref}_1(z'_2)$.

Since ru_i and qv_i are suffix-comparable, $i = 1, 2$, and since g'_1, h'_1 are marked, we can define

$$\overline{g'_1(z)}\,b_1 = ru_1 \wedge_s ru_2 = qv_1 \wedge_s qv_2 = \overline{h'_1(z)}\,b_2 := \tau\,. \tag{20}$$

The definition of the beginning block (b_1, b_2) and the fact, that g'_1 and h'_1 are successors of $\overline{g}$ and $\overline{h}$ imply that τ satisfies (19). The rest follows from the choice of z. □

The previous lemma motivates the following definition. We say that a solution w of I is *long* if a word τ satisfying (19) exists and (w, w) can be decomposed as $(p'_1 w_1 \tau, p'_2 w_2 \tau)$ where $w_1, w_2 \in A^*$ and (p'_1, p'_2) is the beginning block of I. If a solution is not *long*, then we say that it is *short*.

Let us explain how the word "short" is justified. We are interested only in instances which have at most two successors, since we want to deal with branching of the reduction tree. However, there are also harmless successors, namely those which are given by end blocks (u, v) for which ru and qv are not suffix-comparable. These successors admit only solutions which are shorter than $\max\{|p'_1 ru|, |p'_2 qv|\}$ since the pair (r, q) is suffix-comparable with arbitrary product of letter blocks. Consequently, such successors can be seen as leaves of the reduction tree and do not cause its exponential growth.

Therefore, the core of the problem are instances which have at least two end blocks satisfying the assumptions of Lemma 13. For such instances the word τ exists and it is a suffix of all decomposable solutions which are of length at least $\max\{|p'_1 ru|, |p'_2 qv|\}$.

We are now ready to formulate the main result of the paper.

Theorem 14. *Let I have a successor and have two distinct end blocks (u_1, v_1) and (u_2, v_2) such that ru_i and qv_i are suffix-comparable, $i = 1, 2$. Let also $\overline{I}$ have a successor. Let (g_1, h_1) be successor morphisms of (g, h), and suppose that also (g_1, h_1) have a successor. Then q and r are comparable. Define an instance*

$$J = ((p_1', p_2'), g_1, h_1, \mathbf{e}(\varepsilon, q^{-1}r)),$$

where (p_1', p_2') is the beginning block of I. Then I reduces to J in the following sense:

If I has a long solution, then J has a solution. If J has a nonempty solution, then I has a solution.

Proof. By Lemma 10, morphisms $\overline{g_1}$ and $\overline{h_1}$ have successor morphisms. Therefore they have a critical overflow, which implies that $\overline{q}$ and $\overline{r}$ are suffix-comparable (see (5)). Therefore q and r are comparable.

Let w be a long solution of I and let

$$w = p_1' w_1 \tau = p_2' w_2 \tau. \tag{21}$$

From the definition of the beginning block and of τ we have

$$\mathbf{c}(g, h) g(w_1)\, \overline{\mathbf{c}(\,\overline{g}\,,\,\overline{h}\,)} = h(w_2).$$

Equality (12) implies

$$\mathbf{c}(g, h) g(w_1 r) = h(w_2 q)\mathbf{c}(g, h).$$

Thus there is a word w', which satisfies

$$g_1(w') = w_1 r, \qquad\qquad h_1(w') = w_2 q.$$

From (21) we have

$$p_1' g_1(w') r^{-1} \tau = p_2' h_1(w') q^{-1} \tau,$$

and w' is a solution of J.

Let now $w' \in A^+$ be a solution of J, that is,

$$p_1' g_1(w') = p_2' h_1(w') q^{-1} r.$$

By Lemma 11, the word

$$p_1' g_1(w') r^{-1} = p_2' h_1(w') q^{-1}$$

is in A^+. From

$$p_1 g(p_1' g_1(w')) = p_2 h(p_2' h_1(w'))\mathbf{c}(g, h),$$

and equalities (19) and (12) we deduce

$$p_1 g(p_1' g_1(w') r^{-1} \tau) s_1 = p_2 h(p_2' h_1(w') q^{-1} \tau) s_2.$$

Hence

$$w = p_1' g_1(w') r^{-1} \tau = p_2' h_1(w') q^{-1} \tau$$

is a solution of I. □

4 Conclusion

We shall summarize the message of Theorems 12 and 14, and thereby the message of the whole paper.

If we are given an instance $I = ((p_1, p_2), g, h, (s_1, s_2))$ of GPCP, we first check whether I has a short solution.

If there is no short solution and both I and $\overline{I}$ have a successor, then we are interested in the structure of successors that have to be considered in order to decide whether I has a long solution. This is the main topic of this paper. Suppose that I has successors $I'_1, I'_2, \ldots, I'_k$.

If the pair (g_1, h_1) has no successor, then the instances $I'_1, I'_2, \ldots, I'_k$ are leaves of the reduction tree, and their number is bounded by Theorem 12.

If (g_1, h_1) have a successor, then Theorem 14 yields that it is possible to "compress" all successors $I'_1, I'_2, \ldots, I'_k$ into a single instance J to which I reduces. Note that J is not necessarily a successor of I.

Our method allows to construct a reduction tree that has nearly no branching. This structure of the reduction tree strongly indicates that the binary GPCP is in polynomial time. However, to show this explicitly requires many additional details. For example, we have to deal with short solutions or with the situation when only one letter block exists. In particular, it is necessary to consider the depth of the reduction tree. All this goes beyond the scope of the present paper.

Acknowledgements

The work on this paper has been supported by the research project MSM 0021620839.

The authors would like to thank the anonymous referee for careful reading and valuable comments, which helped to improve the presentation.

References

[1] W. S. Brainerd and L. H. Landweber. *Theory of Computation.* John Wiley & Sons, 1974.

[2] C. Choffrut and J. Karhumäki. Combinatorics of Words. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1. pp. 329–438, Springer, 1997.

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

[4] V. Halava. *The Post Correspondence Problem for Marked Morphisms.* PhD thesis, Department of Math., Univ. of Turku. TUCS Dissertations no. 37, 2002.

[5] V. Halava, T. Harju, and M. Hirvensalo. Binary (generalized) Post Correspondence Problem. *Theoret. Comput. Sci.*, 276:183–204, 2002.

[6] V. Halava, T. Harju, and M. Hirvensalo. Generalized Post correspondence problem for marked morphisms. *Internat. J. Algebra Comput.*, 10(6):757–772, 2000.

[7] M. A. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley, 1978.

[8] Š. Holub, Binary equality sets are generated by two words, *J. Algebra*, 259(1): 1–42, 2003.

[9] Š. Holub, A unique structure of two-generated binary equality sets. In M. Ito and M. Toyama, editors, *Developments in Language Theory*, volume 2450 of *Lecture Notes in Computer Science*, pp. 245–257. Springer, 2002.

[10] Š. Holub, Binary morphisms with stable suffix complexity. to appear.

[11] J. E. Hopcroft and J. D. Ullman. *Formal Languages and Their Relation to Automata*. Addison-Wesley, 1969.

[12] Y. Matiyasevich and G. Sénizergues. Decision problems for semi-Thue systems with a few rules. *Theor. Comput. Sci.* 330(1):145-169, 2005.

[13] V. A. Pavlenko. Post combinatorial problem with two pairs of words. *Dokl. Akad. Nauk. Ukr. SSR* 9–11, 1981.

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