The Post Correspondence Problem for Marked Morphisms

by

Vesa Halava

To be presented, with the permission of the Faculty of Mathematics and Natural Sciences of the University of Turku, for public criticism in Auditorium XXI of the University on April 27th, 2002, at 12 noon

University of Turku
Department of Mathematics
FIN-20014 Turku, Finland

2002

Supervisors

Academy Researcher Tero Harju
Department of Mathematics
University of Turku
FIN-20014 Turku
Finland

Professor Juhani Karhumäki
Department of Mathematics
University of Turku
FIN-20014 Turku
Finland

Reviewers

Member of Romanian Academy Gheorghe Păun
Institute of Mathematics of
The Romanian Academy
PO Box 1-764,70700 Bucharest
Romania

Professor Paavo Turakainen
University of Oulu
Department of Mathematical Sciences
Box 3000
FIN - 90014 University of Oulu
Finland

Opponent

Professor Jean Berstel
Institut Gaspard-Monge
Laboratoire d'informatique
Université de Marne-la-Vallée
5 Bd Descartes, Champs-sur-Marne
F-77454 Marne-la-Vallée CEDEX 2
France

ISBN 951-29-2263-0
ISSN 1239-1883
Painosalama Oy

Acknowledgements

First of all, I would like to thank my supervisors Prof. Juhani Karhumäki and Dr. Tero Harju. Juhani originally introduced me to this topic and, as the leader of our research group, he has provided excellent working conditions and atmosphere. Tero I thank especially for the help and suggestions he has given to me during the writing of the thesis. His advice has led to many improvements in the results. I have learned a lot from Tero, not only about mathematics.

Special thanks are due to the co-authors, Ph. Lic. Mika Hirvensalo and Dr. Ronald de Wolf. The papers we wrote together were the kick-off for the thesis.

I would like to thank the Department of Mathematics and Turku Centre for Computer Science (TUCS) for excellent working conditions, and the personnel of both our department and TUCS for the help and support. Especially, I would like to mention my good friend Dr. Lucian Ilie, whose attitude in work I admire.

I also thank the reviewers, Academician Gheorghe Păun and Prof. Paavo Turakainen, for their fruitful comments. The english revision was done by MSc. David Bergen.

Finally, I would like to thank my whole family, especially my wonderful wife Heli for her great support, and my children Vili and Minka for showing me what really is important in life.

April, 2002

Vesa Halava

Contents

Chapter 1

Introduction

A mathematical decision problem consists of a set of *instances*, each of which either has or does not have a certain property, and we are asked to determine whether an instance given as an input has or does not have the property. We say that the problem is *decidable* if there exists an algorithm which for every instance gives the right answer. If no such algorithm exists, then the problem is said to be *undecidable*. It is clear that every well formulated mathematical decision problem is either decidable or undecidable. Unfortunately, there are many significant problems for which we do not yet know whether they are decidable or undecidable. In the above, the existence of an algorithm means, by Church's thesis, that there exists a *Turing machine* which solves the problem.

To prove that a problem is decidable, we must find an algorithm which decides it, or show that such an algorithm exists. The main contributions of this thesis are decidability results, but we also prove a few undecidability results.

The usual method to prove that a problem P is undecidable is to reduce some already known undecidable problem P' to P. This means that we effectively transform an arbitrary instance p' of P' to some instance p of P in such a way that the condition "p has the property of P" is equivalent with the condition "p' has the property of P'". It follows that if the problem P is decidable, then the instances p' of the problem P' can be solved, and therefore the problem P' is decidable, which is a contradiction.

The concept of undecidability plays a major role in contemporary mathematics. In the beginning of the 20th century David Hilbert conjectured that every well-formed mathematical problem is decidable. He also listed the most significant open problems for which the decision procedure should be found. But during the first half of the century Hilbert's dream was destroyed. Surprisingly, some very simply formulated problems were shown to be undecidable. One of these problems is the Post Correspondence Problem, which was proved to be undecidable by Emil Post in 1946 [34]. We

also mention the famous Hilbert's tenth problem, which was proved to be undecidable by Matiyasevich [30].

In the Post Correspondence Problem two lists of words are given, say $\{u_1, \ldots, u_n\}$ and $\{v_1, \ldots, v_n\}$, and the task is to determine whether or not there exists a nonempty sequence $i_1, \ldots i_k$, where $i_j \in \{1, \ldots, n\}$ for $1 \leq j \leq k$, 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 $i_1, \ldots, i_k$ is called a solution of the problem instance. In this thesis we consider the Post Correspondence Problem and its variants in some special cases.

There are two major topics in the theory of computation related to algorithmic decidability questions. The first is to prove new decidability and undecidability results, and the second is to clarify the border between decidability and undecidability. In this thesis we concentrate on these topics. We shall also present proofs for a few recent undecidability results concerning the Post Correspondence Problem.

It is obvious that the special cases of a decidable problem are decidable. On the other hand, a special case of a problem may be decidable eventhough the general problem is undecidable. By searching for decidable cases of undecidable problems, we investigate the borderline between decidability and undecidability. The question is "how much simpler must a problem be made in order to make it decidable?" or to be more precise "what kind of further assumptions have to be imposed to make a problem decidable?"

Here we shall consider the Post Correspondence Problem and its variant in the case where all the words in a list begin with different letters. This variant is called the marked Post Correspondence Problem.

Next we shall shortly consider the structure of the thesis.

In Chapter 2 we give the basic definitions and the notations needed in the thesis. Basic properties of words and languages are also presented in this chapter.

In Chapter 3 we give an introduction to the Post Correspondence Problem (PCP, for short). We redefine the problem by using morphisms and consider briefly the main subproblems and variants of it. We shall prove that the PCP is undecidable for those instances where the morphisms are permutations of each other.

We also define the generalized Post Correspondence Problem, which is the most significant variant of the Post Correspondence Problem studied in the thesis. A proof for the undecidability of the special universe problem concerning the solutions of the generalized Post Correspondence Problem and periodic morphisms is presented.

In Chapter 4 we study the marked Post Correspondence Problem. Our first main result is that this problem is decidable. We shall also prove that it is decidable whether there exists an infinite solution in the case of the marked Post Correspondence Problem. This problem is undecidable for the Post Correspondence Problem in general.

We also prove that the 2-marked Post Correspondence Problem is unde-

cidable. In an instance of this problem all words in the input lists have a different prefix of length two. In this way, a rather good borderline is found between decidability and undecidability.

Note that the marked Post Correspondence Problem is a subproblem of the injective Post Correspondence Problem, which is known to be undecidable.

In Chapter 5 we prove that the generalized Post Correspondence Problem is decidable for the marked instances. Our proof uses the same technique as the proof for the marked Post Correspondence Problem, but it is much more complicated. We also achieve a new proof for the decidability of the binary Post Correspondence Problem, where both of the lists contain only two words. Similarly, the binary generalized Post Correspondence Problem is shown to be decidable.

The material of the thesis comes mostly from the articles [17, 18, 19, 20, 21]. For a short introduction to the topic of the thesis, see [15]. Also, the results from [13] and [16] are presented here.

Chapter 2

Preliminaries

Throughout this work we shall denote the set of the natural numbers by $\mathbb{N} = \{1, 2, \dots\}$ and the set of the integers by $\mathbb{Z}$.

For a set S, we denote by 2^S the *power set* of S, i.e., the collection of all subsets of S.

Sometimes a singleton set $\{u\}$ is written simply as u.

2.1 Basic properties of words

We shall first introduce basic definitions and notations of words. As a general reference to combinatorics on words, we give Choffrut and Karhumäki [4].

Let S be a set and assume that $\circ$ is an associative binary operation on S, that is, for all a, b and c in S,

$$a \circ b \in S \quad \text{and} \quad a \circ (b \circ c) = (a \circ b) \circ c.$$

Then $(S, \circ)$ is called a *semigroup*. Usually, if no confusion is possible, we denote $ab = a \circ b$, and the semigroup $(S, \circ)$ is denoted simply by S.

An element ι in a semigroup S is called an *identity* if for all a in S, $a\iota = \iota a = a$. If there is an identity element in a semigroup S, then S is called a *monoid*. Note that an identity element in a monoid is necessarily unique.

Let A be a finite set of symbols, called an *alphabet*. A *word* over A is a finite sequence of symbols in A. We denote by A^* the set of all words over A. Note that the *empty word* is in A^*. Throughout this work we shall denote the empty word by ε.

Let $u = a_1 \cdots a_n$ and $v = b_1 \cdots b_m$ be two words in A^*, where a_i and b_j are in A for all $1 \leq i \leq n$ and $1 \leq j \leq m$. Define a binary operation $\cdot$ on A^*, called the *concatenation*, by

$$u \cdot v = uv = a_1 \cdots a_n b_1 \cdots b_m.$$

This operation is clearly associative on A^* and, since the empty word is the identity element of A^*, A^* is a monoid. We say that A^* is the *word monoid over* A. The *word semigroup over* A is defined to be $A^+ = A^* \setminus \{\varepsilon\}$.

A semigroup S is called *free* if there is a subset X of S such that each element of S has a unique factorization over X. Such a set X is called a *free generating set of* S, and S is said to be *freely generated by* X. A monoid M with identity element ι is *free* if $M \setminus \{\iota\}$ is a free semigroup.

Note that a word monoid A^* is free and A is its free generating set.

Let A be an alphabet and assume that $u = a_1 \cdots a_n$ is in A^* with $a_i \in A$. The *length* of the word u is n and it is denoted by $|u| = n$.

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 $u \leq v$. A prefix of length k of v is denoted by $\mathrm{pref}_k(v)$ whenever $k \leq |v|$. Set $\mathrm{pref}_k(v) = v$, if $k > |v|$.

Also, if $u \neq \varepsilon$ and $w \neq \varepsilon$ in $v = uw$, then u is a *proper* prefix of v, and this is denoted by $u < v$. We say that u and v are *comparable* if $u \leq v$ or $v \leq u$. 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 $v = wu$. This will be denoted by $u \preccurlyeq v$ and, if $u \neq \varepsilon$ and $w \neq \varepsilon$, then u is called a *proper* suffix of v, denoted by $u \prec v$.

If $u = vw$, then we denote $v = uw^{-1}$ and $w = v^{-1}u$. Note that in the literature "proper" is usually defined without the condition $u \neq \varepsilon$.

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

For an alphabet A, a subset L of A^* is called a *language*.

Finally, we mention that we shall denote the alphabets in this thesis by capital letters.

2.2 Morphisms

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

$$h(uv) = h(u)h(v).$$

It follows that a morphism becomes defined by the images of the letters of the domain alphabet A. Note that for all morphisms h, $h(\varepsilon) = \varepsilon$. If $h(a) \neq \varepsilon$ for all $a \in A$, then h is said to be *nonerasing*.

In this thesis we are mainly interested in marked morphisms. A morphism h is *marked* if it is nonerasing and the images of the letters begin with different letters, i.e., if $h(a) \wedge h(b) = \varepsilon$ for all $a, b \in A$ with $a \neq b$.

A morphism $h\colon A^* \to B^*$ is *injective* if $h(u) \neq h(v)$ for all different $u, v \in A^*$. Clearly, if a morphism is marked, then it is injective.

A morphism $h\colon A^* \to B^*$ is a *prefix* if for all $a, b \in A$, with $a \neq b$, $h(a) \not\leq h(b)$ and $h(b) \not\leq h(a)$, i.e., the images of two different letters are not comparable.

A morphism $h\colon A^* \to B^*$ is called *periodic* if there exists a word $w \in B^*$ such that $h(u) \in w^*$ for all $u \in A^*$.

We say that a morphism $h\colon A^* \to B^*$ is a *coding* if, for all $a \in A$, $h(a) \in B$, that is, if $|h(u)| = |u|$ for all $u \in A^*$.

2.3 Finite automata

We conclude this chapter with the definition of a finite nondeterministic automaton, which is a restricted machine model for accepting languages. As a general reference for the theory of finite automata, we give Eilenberg [7].

Consider a *nondeterministic finite automaton*, NFA for short,

$$\mathcal{A} = (Q, A, T, \sigma, I, F),$$

where Q is the finite set of the *states*, A is the *input alphabet*, T is the alphabet of *edges*, $\sigma\colon T \to Q \times \Sigma \times Q$ is the *transition mapping*, $I \subseteq Q$ is the set of the *initial states* and $F \subseteq Q$ is the set of the *final states*.

The values $\sigma(t)$, for $t \in T$, are the *transitions* of the finite automaton $\mathcal{A}$. An application of a transition $\sigma(t) = (q, a, p)$ (or of the edge t) will change the state q of $\mathcal{A}$ to the state p after reading the input symbol a. We use a set of edges T rather than a transition relation ($\delta \subseteq Q \times \Sigma \times Q$) to allow multiple transitions with the same value, that is, we allow edges t, t' with $t \neq t'$ and $\sigma(t) = \sigma(t')$.

A *path* (or a *computation*) π of $\mathcal{A}$ is a sequence

$$\pi = t_1 t_2 \dots t_n \ (\in T^*),$$

where $t_i \in T$ for all $1 \leq i \leq n$ and $\sigma(t_i) = (q_i, a_i, q_{i+1})$ for some states $q_1, \dots, q_{n+1}$, and input symbols $a_1, \dots, a_n$. Usually we write simply

$$\pi = (q_1, a_1, q_2)(q_2, a_2, q_3) \cdots (q_n, a_n, q_{n+1}). \tag{2.1}$$

The *label* of the path π in (2.1) is the word $\|\pi\| = a_1 \cdots a_n \in A^*$. A path is called *successful* (or *accepting*) and we say that $\mathcal{A}$ *accepts* the word $\|\pi\|$ if $q_1 \in I$ and $q_{n+1} \in F$ in (2.1). A trivial path beginning from $q_0 \in I$ is successful if $q_0 \in F$, and it accepts ε.

The *language accepted* by $\mathcal{A}$ (or the *behaviour* of $\mathcal{A}$) is the subset $L(\mathcal{A})$ of A^* consisting, of the labels of the successful paths of $\mathcal{A}$. Actually, $\|\cdot\|$ can be extended to a morphism from T^* into A^* that becomes defined by $\|t\| = a$, if $\sigma(t) = (q, a, p)$, for all $t \in T$. Then

$$L(\mathcal{A}) = \{w \in A^* \mid w = \|\pi\|,\ \pi \in T^* \text{ is a successful path in } \mathcal{A}\}.$$

An automaton $\mathcal{A}$ is said to be *deterministic* if $\sigma(t_1) = (q, a, p_1)$ and $\sigma(t_2) = (q, a, p_2)$ imply that $t_1 = t_2$ (and therefore also $p_1 = p_2$). This means that for all $q \in Q$ and $a \in A$, there is at most one $p \in Q$ such that (q, a, p) is a transition. We shall also use the notation DFA for a deterministic finite automaton.

Note that usually a nondeterministic finite automaton is defined by using either a transition mapping $\delta\colon Q \times A \to 2^Q$ or a set of transitions $\delta \subseteq Q \times A \times Q$ instead of T and σ. It is clear, however, that our definition of NFA does not change the languages accepted by the finite automata.

A language $L \subseteq A^*$ is called *regular* if there is an NFA $\mathcal{A}$ such that L is accepted by $\mathcal{A}$, i.e., if $L(\mathcal{A}) = L$. Note that each language accepted by an NFA, is also accepted by a DFA. This is a very basic result in the formal language theory (see, e.g., Salomaa [37]).

Let $\mathcal{A} = (Q, A, T, \sigma, I, F)$ be an NFA. There exists another NFA $\mathcal{A}' = (Q \cup \{q_0\}, A, T', \sigma', q_0, F')$ with only one initial state q_0 such that $L(\mathcal{A}') = L(\mathcal{A})$. Such an $\mathcal{A}'$ can be constructed for $\mathcal{A}$ as follows. First define a set C of new edges: for all $t \in T$, if $\sigma(t) = (q, a, p)$ and $q \in I$, then $t' \in C$ such that

$$\sigma'(t') = (q_0, a, p).$$

Finally, let $T' = T \cup C$ and $\sigma'(t) = \sigma(t)$ for all $t \in T$. The set of final states is $F' = F \cup \{q_0\}$ if $\varepsilon \in L(\mathcal{A})$, $F' = F$ otherwise.

There exists a similar procedure for constructing an NFA with only one final state, provided that $\varepsilon \notin L(\mathcal{A})$. This modification is by first defining a new final state and then copying all the transitions ending up in an old final state to this new final state. If $\varepsilon \in L(\mathcal{A})$, then two final states are enough. Indeed, if the initial state is constructed as in the above, there is no transition ending up to the initial state. Therefore, we may set that the initial state is also a final state accepting only ε.

Chapter 3

The Post Correspondence Problem and its variants

In this chapter we shall give an introduction to the Post Correspondence Problem in general. We shall also present proofs for some recent undecidability results concerning this problem.

3.1 Undecidable problem by Emil Post

We already gave one definition of the Post Correspondence Problem in Chapter 1, but we shall now give another equivalent definition.

Let B be an alphabet. In the *Post Correspondence Problem* (PCP for short) we are given a set of n pairs of words,

$$\{(u_i, v_i) \mid u_i, v_i \in B^*,\ i = 1, \ldots, n\},$$

and we are asked to determine whether or not there exists a nonempty sequence $i_1, \ldots, i_k$, where $i_j \in \{1, \ldots, n\}$ for $1 \leq j \leq k$, such that

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

The PCP can be redefined by using morphisms. For this, define two morphisms h and g from A^* into B^*, where $A = \{a_1, \ldots, a_n\}$ is an alphabet of n letters, by

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

Now the original form of the PCP is equivalent to the problem, whether or not there exists a nonempty word $w \in A^+$ such that

$$h(w) = g(w). \tag{3.1}$$

The pair (h, g) is called an *instance* of the PCP. A word w as in (3.1) is called a *solution* of the instance (h, g). The *size* of an instance is the size of the domain alphabet, i.e., the size is equal to $|A|$.

The following result was proved by Post [34] in 1946.

Theorem 3.1. *The PCP is undecidable, i.e., there does not exist an algorithm, which, for a given input instance (h, g) (of any size) tells whether or not the instance has a solution.*

Theorem 3.1 follows from the undecidability of the halting problem for Turing machines, the reason being that two morphisms together can simulate the computations of a given Turing machine on a given input (see Rozenberg and Salomaa [35]).

The PCP is one of the most important undecidable problems, not only in combinatorics on words, because it is very useful in proving other undecidability results. As examples of such results we mention some decidability issues in the theory of matrices. A clever coding technique introduced by Paterson [32] defines a strong connection between the PCP and certain 3×3 integer matrices. For example, it can be proved that for a finite set of 3×3 integer matrices $M_1, M_2, \ldots, M_n$, it is undecidable whether the zero matrix belongs to the semigroup generated by the matrices M_i (see Paterson [32] and Halava and Harju [14]). There are also many other undecidability results in matrix theory that are proved by using the PCP (see Halava [10] or Harju and Karhumäki [22]).

We shall consider some known restrictions of the PCP. This we do in order to illustrate the proofs for decidability and undecidability and at the same time present some recent results on the PCP. Before that we shall define the most significant modification of the PCP in this thesis.

3.2 The generalized Post Correspondence Problem

An instance of the *generalized Post Correspondence Problem* (*GPCP*, for short) consists of two morphisms $h, g\colon A^* \to B^*$ 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

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

The word w is again called a *solution* of the instance $((p_1, p_2), h, g, (s_1, s_2))$ of the GPCP. The next theorem is obvious, since all instances of the PCP are also instances of the GPCP.

Theorem 3.2. *The GPCP is undecidable.*

From this point on we also allow the empty word to be a solution of an instance of the GPCP. Note that this does not affect the undecidability of the GPCP, although the instances of the PCP become trivial. This convention is adopted in order to simplify the notations.

In the theory of computability we are not only interested in proving problems to be decidable or undecidable. Also, restricting an undecidable problem by making some further assumptions or making other modifications

may make the problem decidable. In this way we investigate the "borderline" of decidability and undecidability. Next we shall harvest this point of view.

3.3 Restricting the morphisms

In the PCP, we may set further assumptions on the two morphisms that define an instance. For example, if we assume that the size of an instance of the PCP is two, i.e., the domain alphabet A is binary, then the PCP is decidable. This was originally proved by Ehrenfeucht, Karhumäki and Rozenberg [5] and Pavlenko [33]. In this thesis we achieve a simpler and shorter proof for this result. This problem is considered in details in Section 5.4. On the other hand, if $n \geq 7$, then the PCP remains undecidable (see Matiyasevich and Sénizergues [31]). Therefore we state

Theorem 3.3. *The PCP is decidable when $|A| \leq 2$, and it is undecidable when $|A| \geq 7$.*

Problem 3.1. *Is the PCP decidable for instances of size n, when $3 \leq n \leq 6$?*

Since Problem 3.1 is open, the borderline of decidability and undecidability is somewhere between two and seven, when considering the alphabet size.

Next we shall consider the injective morphisms. The following theorem was proved by Lecerf [27].

Theorem 3.4. *The PCP is undecidable for those instances where the morphisms are injective.*

Our main interest in this thesis is to consider the restriction of the PCP where h and g are marked (Chapter 4). Clearly, if a morphism is marked, then it is injective. The converse does not hold. Consider, for instance, $A = B = \{1, 2\}$, $g(1) = 11$, $g(2) = 12$, then g is injective, but not marked.

In the *marked Post Correspondence Problem* we assume that in each instance (h, g) the morphisms h and g are marked. We shall prove in Chapter 4 that the marked PCP is decidable for alphabets of any size. This result was proved by Halava, Hirvensalo and de Wolf [21].

The following bounds are known for the GPCP.

Theorem 3.5. *The GPCP is decidable when $|A| \leq 2$, and it is undecidable when $|A| \geq 7$.*

In the previous theorem, the decidability result was originally proved by Ehrenfeucht, Karhumäki and Rozenberg [5]. We shall present a new proof in Chapter 5. The undecidability result was proved by Harju, Karhumäki and Krob [23]. As in the case of the PCP, we state the following open problem.

Problem 3.2. *Is the GPCP decidable for instances of size n, when $3 \leq n \leq 6$?*

In the *marked* GPCP we assume that in each instance the morphisms are marked. The motivation for studying the marked PCP and the marked GPCP originates from [5], where Ehrenfeucht, Karhumäki and Rozenberg proved that the binary PCP is decidable. They actually proved that every instance of the binary PCP is either periodic or it can be reduced to an equivalent instance of the marked GPCP with binary alphabets, and then they showed by case analysis that the binary marked GPCP is decidable.

It was proved by Halava, Harju and Hirvensalo [19] that the marked GPCP is decidable in the general setting, that is, regardless of the size of the domain alphabet. Although the proof uses the same basic idea as the proof of the decidability of the marked PCP by Halava, Hirvensalo and de Wolf [21], the technical parts involved in the generalization turned out to be considerably more involved. The proof of the decidability of the marked GPCP will be presented in Chapter 5.

3.3.1 Decidability of the periodic GPCP and PCP

We shall begin with a simple decidability result for the periodic morphisms. The PCP becomes decidable if one of the morphisms is periodic. We shall call an instance of the PCP *periodic* if one of the morphisms is periodic.

The decidability of the periodic PCP will be used in the proof for the decidability of the binary PCP. We shall next prove that the GPCP is decidable for those cases where one of the morphisms is periodic and the decidability of the periodic PCP follows from this result. The proof uses the idea of proof for the decidability of the periodic PCP in Harju and Karhumäki [22] (see also Ehrenfeucht, Karhumäki and Rozenberg [5]). We use properties of one-counter, or actually context-free languages (see for example Berstel [2] or Salomaa [37]).

Theorem 3.6. *The GPCP is decidable for instances where one of the morphisms is periodic.*

Proof. Assume that $((p_1, p_2), h, g, (s_1, s_2))$ is an instance of the GPCP, where $h, g\colon A^* \to B^*$, such that h is periodic, i.e., $h(A^*) \subseteq u^*$ for a word $u \in B^*$. It is easy to show that the language

$$L = \{v \mid |h(v)| + |p_1 s_1| = |g(v)| + |p_2 s_2|\},$$

is a one-counter language and therefore context-free, and $R = g^{-1}(R_0)$, where $R_0 = p_2^{-1}(p_1 u^* s_1) s_2^{-1}$, is a regular language. Now $w \in L \cap R$ if and only if $g(w) \in R_0$ and $|p_2 g(w) s_2| = |p_1 h(w) s_1|$. In other words, we have $p_2 g(w) s_2 = p_1 h(w) s_1$ for some w if and only if $L \cap R \neq \emptyset$. The claim follows since $L \cap R$ is context-free and the emptiness problem is decidable for context-free languages (see for example Salomaa [37]). □

Theorem 3.7. *The PCP is decidable for instances (h, g) where h is periodic.*

Proof. As mentioned before, the PCP is a special case of the GPCP if we exclude the trivial solution ε. Therefore, the proof goes as the proof of Theorem 3.6 if we define $R = g^{-1}(R_0) \setminus \{\varepsilon\}$.

□

3.3.2 Undecidability result for permuted morphisms

We shall next show that the PCP remains undecidable for those instances (h, g) where the second morphism is a permutation of the first one. Recall that a mapping π from the set A onto A is a permutation if it is bijective. For a morphism $h\colon A^* \to B^*$ and a permutation π of the set A, we shall denote by $h\pi$ the *permutation of h by π*, that is, for all $a \in A$,

$$h\pi(a) = h(\pi(a)).$$

In the proof of Theorem 3.8 we use the undecidability of the GPCP. The next theorem was proved by Halava and Harju [15].

Theorem 3.8. *The PCP is undecidable for instances $(h, h\pi)$ where $h\colon A^* \to B^*$ is a morphism and $\pi\colon A \to A$ is a permutation.*

Proof. Assume that $((p_1, p_2), h, g, (s_1, s_2))$ is an instance of the GPCP, where $h, g\colon A^* \to B^*$ and $A = \{a_1, \ldots, a_n\}$.

Let $\overline{A} = \{\overline{a}_1, \ldots, \overline{a}_n\}$ with $\overline{A} \cap A = \emptyset$, and let a, b, d be new letters. Let ℓ and r be two morphisms from B^* to $(B \cup \{d\})^*$ such that for all $x \in B$,

$$\ell(x) = dx \quad \text{and} \quad r(x) = xd.$$

Now let $h'\colon (A \cup \overline{A} \cup \{a, b, \overline{a}, \overline{b}\})^* \to (B \cup \{a, b, d\})^*$ be defined by

$$\begin{aligned} h'(a) &= a \cdot \ell(p_1), & h'(\overline{a}) &= ad \cdot r(p_2), \\ h'(a_i) &= \ell(h(a_i)), & h'(\overline{a}_i) &= r(g(a_i)) \quad \text{for } i = 1, 2, \ldots, n, \\ h'(b) &= \ell(s_1) \cdot db, & h'(\overline{b}) &= r(s_2) \cdot b. \end{aligned}$$

Consider the permutation π of $A \cup \overline{A} \cup \{a, b, \overline{a}, \overline{b}\}$ given by $\pi(x) = \overline{x}$ and $\pi(\overline{x}) = x$ for all $x \in A \cup \{a, b\}$. It is straightforward to show that the instance $(h', h'\pi)$ has a solution if and only if the instance $((p_1, p_2), h, g, (s_1, s_2))$ of the GPCP has a solution. Indeed, the solutions of the instance $(h', h'\pi)$ are necessarily in the set

$$\left(aA^*b \ \cup \ \overline{a}\overline{A}^*\overline{b}\right)^*,$$

because of the marker letters a and b, and because of the "unsynchronized" morphisms ℓ and r. Consequently, the solutions of minimal length are in aA^*b or in $\overline{a}\overline{A}^*\overline{b}$.

Clearly, a word $\overline{w} \in \overline{a}\overline{A}^*\overline{b}$ is a solution to $(h', h'\pi)$ if and only if $w = \pi(\overline{w}) \in aA^*b$ is a solution, where w is obtained from $\overline{w}$ by removing the bars from the letters. In the case $w = avb$ with $v \in A^*$,

$$h'(w) = a \cdot \ell(p_1 h(v) s_1) \cdot db,$$
$$h'\pi(w) = ad \cdot r(p_2 g(v) s_2) \cdot b = a \cdot \ell(p_2 g(v) s_2) \cdot db,$$

and therefore $h'(w) = h'\pi(w)$ if and only if $p_1 h(v) s_1 = p_2 g(v) s_2$, since ℓ is injective. This proves the claim. □

Similar ideas as in the previous proof can be used to prove other undecidable variants of the PCP (see especially Harju and Karhumäki [22]).

3.4 Modifications of the problem

In the previous sections we considered mainly the computability issue in the case of the PCP, where the morphisms were restricted somehow. In this section we shall modify the problem itself. Actually, we already considered the GPCP, which is one of such problems.

3.4.1 Prefix PCP

We shall state a special undecidability result for the prefix property of the images of two morphisms. The proof can be found from Halava [11] (see also Halava and Harju [12]). In the proof, the halting problem of the Turing machine is reduced to the problem.

Theorem 3.9. *Let B be an alphabet, and a and c be letters not in B. It is undecidable, for instances (h, g) of two nonerasing morphisms, whether there exists a word $w \in a^+B^*c$ such that $g(w)$ and $h(w)$ are comparable, where $h(a) \in g(a)^*$, and $|g(x)| = |h(x)|$ for all $x \in B \cup \{c\}$.*

Note that the previous theorem does not imply that it is undecidable for morphisms $h, g\colon A^* \to B^*$, whether there exists a word $w \in A^*$ such that $g(w) < h(w)$. In the theorem the prefix property is rather special.

The existence of a word w such that $g(w)$ and $h(w)$ are comparable is trivially decidable for the morphisms h and g, as in Theorem 3.9, since if $g(w)$ and $h(w)$ are comparable for some w, then already $g(a)$ and $h(a)$ are comparable for at least one letter a. Therefore, the fixed ending with c is essential in Theorem 3.9.

Note that the problem whether $g(w) < h(w)$ for some word w is undecidable for arbitrary morphisms. This result follows from the proof of the undecidability of the PCP using the halting problem. On the other hand, if the order of the morphisms is not fixed then the problem becomes trivially decidable as discussed in the above.

3.4.2 Special universe problem for GPCP

For an instance $I = (h, g)$ of the PCP, let

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

be its *equality set*. Note that the equality set is the set of all solutions of the instance (h, g) including the *trivial solution* ε.

It is known that the equality set is a complement of a one-counter language (see Harju and Karhumäki [22] and Rozenberg and Salomaa [35], also for further properties of the equality sets). Various properties of the solutions are also studied in Lipponen [28].

Similarly, for an instance $J = ((p_1, p_2), h, g, (s_1, s_2))$ of the GPCP, we shall define

$$E_G(J) = \{w \in A^* \mid p_1 h(w) s_1 = p_2 g(w) s_2\}$$

as the *generalized equality set* of J.

In this section we shall consider a special decision problem concerning the equality sets of the instances of the GPCP by restricting ourselves to the instances where the second morphism g is periodic. The result is from Halava and Harju [16].

In this problem we are given an instance $J = ((p_1, p_2), h, g, (s_1, s_2))$, where $h, g\colon A^* \to B^*$, and a coding $\mu\colon A^* \to C^*$, and it is asked whether or not the equality

$$\mu(E_G(J)) = C^+, \tag{3.2}$$

holds. Recall that a morphism μ is a coding if for all $a \in A$, $h(a) \in C$.

Note that the same problem for the equality sets $E(I)$ (of the PCP) is trivially decidable. Actually, we prove that the universe problem is undecidable for the simple instances of the form

$$J_a = ((\varepsilon, a), h, g, (\varepsilon, \varepsilon)), \tag{3.3}$$

where $a \in B$ is a fixed letter, and g is a periodic morphism. Recall also that the (G)PCP is decidable for those instances where the other morphism is periodic (see Theorems 3.6 and 3.7).

In the proof of the undecidability of the property (3.2) we shall use a result from [12], where it is shown that the universe problem is undecidable for the integer weighted finite automata.

Let $\mathbb{Z}$ be the additive group of the integers. An *integer weighted finite automaton* $\mathcal{A}^\gamma$ consists of a finite automaton $\mathcal{A} = (Q, C, T, \sigma, q_0, F)$, and a *weight function* $\gamma\colon T \to \mathbb{Z}$ of the edges. To simplify notations, we shall write the edges in the form

$$t = \langle q, a, p, z \rangle\,,$$

if $\sigma(t) = (q, a, p)$ and $\gamma(t) = z$. Similarly, we shall write the transition mapping σ as a set, $\sigma \subseteq Q \times C \times Q \times \mathbb{Z}$, where

$$\sigma = \{\langle q, a, p, z\rangle \mid \exists t \in T\colon \sigma(t) = (q, a, p) \text{ and } \gamma(t) = z\}.$$

Note that the weight function γ does not affect the computations of the original finite automaton $\mathcal{A}$.

The family of all finite automata with integer weights is denoted by $\mathrm{FA}(\mathbb{Z})$.

Let $\mathcal{A} \in \mathrm{FA}(\mathbb{Z})$, and let $\pi = t_0 t_1 \cdots t_n$ be a path of $\mathcal{A}$ where $\sigma(t_j) = (q_j, a_j, q_{j+1})$ for $0 \leq j \leq n$. The *weight of the path* π is the integer

$$\gamma(\pi) = \sum_{i=0}^{n} \gamma(t_i).$$

Define $L(\mathcal{A}^\gamma) = \|\gamma^{-1}(0)\|$, that is,

$$L(\mathcal{A}^\gamma) = \{w \in C^* \mid w = \|\pi\|,\ \gamma(\pi) = 0 \text{ and } \pi \text{ is a successful path}\},$$

to be the language *accepted by* $\mathcal{A}^\gamma$.

We note that the model of the integer weighted finite automaton defined above is similar to the blind counter machine introduced by Greibach [9] and closely related to the defense systems introduced by Lisovik in [29] . It is also closely related to the 1-turn counter automata as considered by Baker and Book [1], Greibach [8], and especially by Ibarra [26]. In the present model the counter is replaced by a weight function of the transitions, and while doing so, the finite automaton becomes independent of the counter.

The next lemma was proved by Halava and Harju [12].

Lemma 3.1. *It is undecidable for automata* $\mathcal{A}^\gamma \in \mathrm{FA}(\mathbb{Z})$ *whether or not* $L(\mathcal{A}^\gamma) = C^*$.

Actually, a much stronger result was proved in [12]. In particular, it was shown that the universe problem $L(\mathcal{A}^\gamma) = C^*$ is undecidable for 4-state unimodal integer weighted finite automata where all states are final. An $\mathcal{A}^\gamma \in \mathrm{FA}(\mathbb{Z})$ is said to be *unimodal* if for all paths in $\mathcal{A}^\gamma$ (not only for the accepting paths of weight 0) the weights of the transitions are first strictly positive, then 0, then strictly negative and finally 0.

Note that we may assume by Section 2.3 that the automata in Lemma 3.1 have only one final state and, furthermore, that there are no edges leaving the unique final state. Therefore we state that

Corollary 3.1. *It is undecidable for automata* $\mathcal{A}^\gamma \in \mathrm{FA}(\mathbb{Z})$, *where there are no transitions to the initial state and no transitions from the final state, whether or not* $L(\mathcal{A}^\gamma) = C^+$.

We begin with a simple decidability property.

Theorem 3.10. *Let $I = (h, g)$ with $h, g\colon A^* \to B^*$ be an instance of the PCP, and let $\mu\colon A^* \to C^*$ be a coding. It is decidable whether or not $\mu(E(I)) = C^*$.*

Proof. We have $\mu(E(I)) = C^*$ if and only if for each letter $a \in C$ the set $\mu^{-1}(a) \cap E(I)$ is nonempty. Indeed, if $u, v \in E(I)$, also $uv \in E(I)$. Since the sets $\mu^{-1}(a)$ are finite, and the membership problem (that is, given a word w, determine whether $w \in E(I)$) is decidable for all instances, the claim follows. □

We shall now prove the main theorem of this section.

Theorem 3.11. *It is undecidable whether or not $\mu(E_G(J)) = C^+$ for codings μ, and instances $J_\# = ((\varepsilon, \#), h, g, (\varepsilon, \varepsilon))$, where g is periodic and $\#$ is a letter.*

Proof. Let $\mathcal{A}^\gamma = (Q, C, T, \sigma, q_0, q_n) \in \mathrm{FA}(\mathbb{Z})$ be as in Corollary 3.1, where $Q = \{q_0, q_1, \ldots, q_n\}$. Especially, we require that there are no transitions to q_0 or from q_n.

Define the alphabets $A = \{[i, a, j, z] \mid \langle q_i, a, q_j, z\rangle \in \sigma\}$ and $B = \{c, d, \#\}$, and two mappings $+$ and $-$ for $z \in \mathbb{Z}$ by

$$z_+ = \begin{cases} z & \text{if } z > 0, \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \quad z_- = \begin{cases} 0 & \text{if } z > 0, \\ |z| & \text{otherwise.} \end{cases}$$

Let

$$\alpha_\ell = \begin{cases} d^{n-\ell-1}c, & \text{if } \ell \geq 1, \\ \#d^n c, & \text{if } \ell = 0, \end{cases} \quad \text{and} \quad \beta_\ell = \begin{cases} d^{\ell+1}, & \text{if } \ell \leq n-1, \\ \varepsilon, & \text{if } \ell = n, \end{cases}$$

for $\ell = 0, \ldots, n$. The morphisms h, g and μ are defined as follows for each $[i, a, j, z] \in A$:

$$h([i, a, j, z]) = \alpha_i (d^n c)^{z_+} \beta_j$$

and

$$g([i, a, j, z]) = (d^n c)^{z_- + 1} \quad \text{and} \quad \mu([i, a, j, z]) = a.$$

In particular, the morphism g is periodic and μ is a coding.

Denote $J = ((\varepsilon, \#), h, g, (\varepsilon, \varepsilon))$. We shall prove that $\mu(E_G(J)) = L(\mathcal{A}^\gamma)$.

Assume first that $x = a_1 \cdots a_k \in \mu(E_G(J))$, where $a_\ell \in C$ for all $1 \leq \ell \leq k$. Then there exists a word $w \in \mu^{-1}(x) \subseteq A^*$,

$$w = [i_1, a_1, j_1, z_1][i_2, a_2, j_2, z_2] \cdots [i_k, a_k, j_k, z_k],$$

such that $w \in E_G(J)$. Consequently, $h(w) = \# g(w)$, where

$$g(w) = (d^n c)^{\sum_{\ell=1}^{k} (z_\ell)_- + k}. \tag{3.4}$$

Since $h(w)$ begins with the letter $\#$, necessarily $i_1 = 0$. Similarly, since $g(w)$ ends with $d^n c$, and hence also $h(w)$, necessarily $j_k = n$.

Consider next the images of h of two consecutive letters in w. Such an image is of the form

$$\alpha_{i_\ell}(d^n c)^{(z_\ell)_+}\beta_{j_\ell}\alpha_{i_{\ell+1}}(d^n c)^{(z_{\ell+1})_+}\beta_{j_{\ell+1}},$$

for some $1 \leq \ell \leq n-1$. In order for $\beta_{j_\ell}\alpha_{i_{\ell+1}} = d^n c$ to hold, necessarily $j_\ell = i_{\ell+1}$ and $i_{\ell+1}, j_\ell \notin \{0, n\}$. Hence, we have that

$$w = [0, a_1, j_1, z_1][j_1, a_2, j_2, z_2]\cdots[j_{k-1}, a_k, n, z_k],$$

and, therefore, $h(w) = \#(d^n c)^{\sum_{\ell=1}^{k}(z_\ell)_+ + k}$. On the other hand, the equality $h(w) = \#g(w)$ implies, by (3.4),

$$\sum_{\ell=1}^{k}(z_\ell)_+ = \sum_{\ell=1}^{k}(z_\ell)_-,$$

and, therefore

$$\sum_{\ell=1}^{k} z_\ell = \sum_{\ell=1}^{k}(z_\ell)_+ - \sum_{\ell=1}^{k}(z_\ell)_- = 0.$$

By the definition of A, we obtain that there exists an accepting path for the word $a_1 a_2 \cdots a_n = \mu(w) = x$ in $\mathcal{A}^\gamma$. It follows that $\mu(E_G(J)) \subseteq L(\mathcal{A}^\gamma)$.

Assume next that $x = a_1 \cdots a_k \in L(\mathcal{A}^\gamma)$, where $a_i \in C$ for all $1 \leq i \leq k$. Then there exists a sequence

$$\langle q_{j_{i-1}}, a_i, q_{j_i}, z_i\rangle \in \sigma$$

of transitions where $1 \leq i \leq k$, $j_i \in \{0, \ldots, n\}$, and $j_0 = 0$ and $j_k = n$, such that $\sum_{i=1}^{k} z_i = 0$. Clearly,

$$w = [0, a_1, j_1, z_1][j_1, a_2, j_2, z_2]\cdots[j_{k-1}, a_k, n, z_k] \in \mu^{-1}(x).$$

By the definitions of h and g,

$$h(w) = \#(d^n c)^{\sum_{\ell=1}^{k}(z_\ell)_+ + k} \quad \text{and} \quad g(w) = (d^n c)^{\sum_{\ell=1}^{k}(z_\ell)_- + k}.$$

Since $\sum_{i=1}^{k} z_i = 0$, we have that

$$\sum_{i=1}^{k}(z_i)_+ = \sum_{i=1}^{k}(z_i)_-,$$

which implies $h(w) = \#g(w)$, i.e., $w \in E_G(J)$. Moreover, since $\mu(w) = x$, we obtain that $x \in \mu(E_G(J))$. It follows that $L(\mathcal{A}^\gamma) \subseteq \mu(E_G(J))$.

Because $\mu(E_G(J)) = L(\mathcal{A}^\gamma)$, we have that $\mu(E_G(J)) = C^+$ if and only if $L(\mathcal{A}^\gamma) = C^+$, and the claim follows from the undecidability result in Corollary 3.1. □

Note that in the above proof the special symbol # is used only to force the transition sequences in $\mathcal{A}^\gamma$ simulated by morphisms to start from the initial state. It also guarantees that concatenations of such sequences are not allowed in $E_G(J)$.

Note also that in Theorem 3.10 we did not restrict ourselves to the periodic instances of the PCP as in Theorem 3.11. In the case of periodic instances, the decidability result in Theorem 3.10 is true for all morphisms μ, not only for codings. This follows because the PCP is decidable in the periodic case.

Note that here we also find a borderline between decidability and undecidability in the sense that a decidable problem turns into an undecidable one by adding one single letter, #.

Chapter 4

The marked Post Correspondence Problem

In this chapter we shall study the marked PCP, defined in Section 3.3. Recall that an instance of the PCP is marked if the morphisms are marked, that is, the images of the letters in each morphism begin with different letters.

4.1 Motivation

The proof of the decidability of the binary PCP in Ehrenfeucht, Karhumäki and Rozenberg [5] is based on a reduction where an arbitrary instance of the binary PCP is transformed into an equivalent instance of the binary generalized PCP, where the morphisms are marked. Here the equivalence between the instances means that one of the instances has a solution if and only if they both have a solution. In this chapter we shall consider the marked PCP and prove that it is decidable for alphabets of any size. This result is the first of our main results in this thesis, and it is also crucial in the proof of the decidability of the marked generalized PCP in Chapter 5. In fact, we shall show that the marked PCP is in **EXPTIME** (the class of languages that can be recognized in time bounded by $2^{p(N)}$ for some polynomial p in the input size N) and in **PSPACE** (the class of languages that can be recognized using space bounded by $p(N)$ for some polynomial p in the input size N).

As stated before, the PCP can be used to establish the borderline between decidability and undecidability. The main result of this chapter is the decidability of marked PCP. In Section 4.3 we extend the notation of marked morphism to k-marked morphism. We say that a morphism h is *k-marked* if for all letters a, b, $\text{pref}_k(h(a)) \neq \text{pref}_k(h(b))$. Note that, by definition, if $|h(a)| < k$, then $\text{pref}_k(h(a)) = h(a)$. We shall prove that the PCP for 2-marked morphisms is undecidable, thus locating the decidability / undecidability borderline between 1-marked and 2-marked instances.

In another direction, we can weaken the assumption that the morphisms are marked by requiring g and h to be prefix or even *biprefix* morphisms. Recall that g is a prefix morphism if $g(a)$ is not a prefix of $g(b)$, for all $a \neq b$, and a biprefix morphism if, for all letters a and b, $g(a)$ is not a prefix nor suffix of $g(b)$ whenever $a \neq b$. The following theorem was proved by Ruohonen [36].

Theorem 4.1. *The PCP is undecidable for biprefix morphisms. In particular, the PCP is undecidable for prefix morphisms.*

Note that each marked morphism is a prefix morphism. Both marked and prefix morphisms are special cases of injective morphisms.

4.1.1 Basic properties

The marked morphisms are "deterministic" in the sense that for a given word w, there is at most one letter such that the image of the letter is a prefix of w. We illustrate this in the following lemma, but first we give a definition of a g-cover. Let $g\colon A^* \to B^*$ be a marked morphism. A word $u \in A^*$ such that

$$w \leq g(u) \quad \text{and } g(u') < w, \quad \text{for all } u' < u,$$

is called a *g-cover of w*.

Lemma 4.1. *Let $g\colon A^* \to B^*$ be a marked morphism and $w \in B^*$ be a nonempty word. If w has a g-cover, then it is unique.*

Proof. We prove the claim by induction on the length of w. First, if $|w| = 1$, then since g is marked there exists at most one letter a in A such that $g(a)$ begins with w. Assume now that the claim is true for all words of length k.

Let $|w| = k + 1$ and assume, contrary to the claim, that there exist two words, u_1 and u_2, which fulfill the condition of the claim, and $u_1 \neq u_2$. Now since g is marked, there exists at most one letter $a \in A$ such that $g(a) \wedge w \neq \varepsilon$. Then clearly $a < u_i$ for $i = 1, 2$ and the word $g(a)^{-1}w$ has two different g-covers, $a^{-1}u_1$ and $a^{-1}u_2$, but this contradicts the induction hypothesis. □

Corollary 4.1. *Let $h, g\colon A^* \to B^*$ be two marked morphisms. For a word $w \in B^+$, there exists at most one pair (u, v) of words $u, v \in A^*$ such that*

$$wh(u) = g(v)$$

and $wh(u') \neq g(v')$ for all $u' \leq u$ and $v' < v$.

Proof. Assume, contrary to the claim, that there exist two different such pairs (u_1, v_1) and (u_2, v_2). Now $u_1 = u_2$ if and only if $v_1 = v_2$, since h and g are marked. We can thus assume that $u_1 \neq u_2$ and $v_1 \neq v_2$.

Let $v' = v_1 \wedge v_2$. Clearly v' is nonempty, since w has a unique g-cover by Lemma 4.1. Next the word $w_0 = w^{-1}g(v')$ has a unique h-cover u', unless $g(v') = w$, in which case $u_1 = \varepsilon = u_2$; a contradiction. Therefore $v' < v_i$ and $u' \leq u_i$ for $i = 1, 2$. Now the word $g(v')^{-1}wh(u') \neq \varepsilon$ has a unique g-cover $v'' \neq \varepsilon$ and clearly $v'v'' < v_1$ and $v'v'' < v_2$. This contradicts the assumption that v' is the maximal common prefix of v_1 and v_2. □

4.1.2 Blocks

We aim at a decision method for the marked PCP. To begin with, we start with the following simpler problem:

> Given an instance $I = (h, g)$ of the marked PCP, where $h, g\colon A^* \to B^*$, and $a \in B$. Does there exist $x, y \in A^+$ such that $h(x) = g(y)$ and $a \leq h(x)$?

We do not look for solutions for $h(x) = g(x)$ here, but only for $h(x) = g(y)$, and we additionally require that $h(x)$ begins with a specific letter a. This problem is known to be decidable for two morphisms in general, the reasoning being that the language $h(A^*) \cap aB^*$ is regular, and there exist such words x and y if and only if

$$\big(h(A^*) \cap aB^*\big) \cap \big(g(A^*) \cap aB^*\big) \neq \emptyset, \tag{4.1}$$

and the emptiness problem is decidable for regular languages.

Example 4.1. Let $I = (h, g)$ be a marked instance such that

$$\begin{array}{llll} g(a_1) = a_1, & g(a_2) = a_2, & g(a_3) = a_3a_4, & g(a_4) = a_4, \\ h(a_1) = a_1a_3, & h(a_2) = a_4a_2, & h(a_3) = a_3a_3, & h(a_4) = a_2a_2. \end{array}$$

Then, for $a = a_1$, a solution of the above problem is $y = a_1a_3a_2$ and $x = a_1a_2$.

If $h(u) = g(v)$ and $h(u') \neq g(v')$ for all $u' < u$ and $v' < v$, then the pair (u, v) is called a *minimal solution* to the equation $h(x) = g(y)$.

Lemma 4.2. *Let h and g be marked morphisms, where $h, g\colon A^* \to B^*$, and let $a \in B$. There exists at most one minimal solution (u, v) to the equation $h(x) = g(y)$ such that $a \leq h(u) \wedge g(v)$. Moreover, such a minimal solution for a given letter a can be effectively found.*

Proof. Let $b \in A$ be a letter such that $a \leq h(b)$. Then b is unique with this property. By Corollary 4.1, the word $h(b)$ has at most one pair (u_1, v_1) such that $h(b)h(u_1) = g(v_1)$ and $h(b)h(u') \neq g(v')$ for all $u' \leq u_1$ and $v' < v_1$. Clearly, if such a pair (u_1, v_1) exists, then $(u, v) = (bu_1, v_1)$ is a minimal solution and it is the only one by the uniqueness of b and (u_1, v_1).

The last part of the claim follows easily, since we can find a minimal solution by exhaustive search (see (4.1)). □

The minimal solution (u, v) of the equation $h(x) = g(y)$ under the constraint $a \leq h(x) \wedge g(y)$ is called a *block for the letter* a. We shall denote this by $\beta(a) = (u, v)$ if the block (i.e., the minimal solution) exists. Otherwise, $\beta(a)$ is not defined. Furthermore, if $\beta(a)$ is defined, then a is called a *block letter*.

Assume that $w \in A^+$ is a solution of the instance (h, g) of the marked PCP. It is clear that there exists a unique *block decomposition* of w,

$$w = u_1 u_2 \cdots u_k = v_1 v_2 \cdots v_k, \tag{4.2}$$

where, $k \geq 1$, $(u_i, v_i) = \beta(a_i)$ and $a_i \in A$ for all $i = 1, \ldots, k$. In particular, each solution is a concatenation of blocks.

$h(w)$	$h(u_1)$	$h(u_2)$	$\cdots$	$h(u_k)$
$g(w)$	$g(v_1)$	$g(v_2)$	$\cdots$	$g(v_k)$

Figure 4.1: A block decomposition of a solution w

4.2 Decidability of the marked PCP

In this section we shall construct an algorithm for the marked PCP. To this end we use the blocks iteratively to compress the solutions.

4.2.1 Successors

Let (h, g) be an instance of the marked PCP for $h, g\colon A^* \to B^*$. We make two assumptions: firstly, $A \subseteq B$ and secondly,

$$a \leq h(a) \quad \text{for all } a \in A. \tag{4.3}$$

The first assumption is achieved by replacing B with $A \cup B$ and the second one is achieved by applying a suitable permutation to the images of h and g. *From this point on we shall assume that all instances of the marked PCP fulfill these two conditions.*

By using blocks we shall define for an instance (h, g) its *successor* (h', g'). Let

$$A' = \{a \mid \beta(a) \text{ exists } \} \subseteq A (\subseteq B).$$

We define the morphisms $h', g'\colon (A')^* \to A^*$ by

$$h'(a) = u \quad \text{and} \quad g'(a) = v,$$

if $\beta(a) = (u, v)$.

Lemma 4.3. *The morphisms h' and g' are marked.*

Proof. The claim follows by the definition of a block: the morphisms h and g are marked and for each letter in B, there is at most one block. □

By the previous lemma, the successor (h', g') of an instance (h, g) is also an instance of the marked PCP. We shall next show that an instance and its successor are equivalent, i.e., (h, g) has a solution if and only if (h', g') has a solution.

Lemma 4.4. *Let $I = (h, g)$ be an instance of the marked PCP and $I' = (h', g')$ be its successor. Then*

(i) $hh'(x) = gg'(x)$ *for all* $x \in (A')^*$.

(ii) If w is a solution of I, then there is a solution w' for I' such that $w = h'(w')$.

(iii) If w' is a solution of I', then $h'(w')$ $(= g'(w'))$ is a solution of I.

(iv) $a \leq h'(a)$ *for each letter* $a \in A'$.

Proof. The first claim follows from the definition of a block: for a letter $a \in A'$, let $\beta(a) = (u, v)$. Then $h'(a) = u$ and $g'(a) = v$, and therefore $h(h'(a)) = h(u) = g(v) = g(g'(a))$. Since hh' and gg' are morphisms, the claim follows.

For (ii), assume that I has a solution $w = u_1 \cdots u_k = v_1 \cdots v_k$, where $(u_i, v_i) = \beta(a_i)$ for the letters $a_i \in A'$, $1 \leq i \leq k$. By the definition of h' and g',

$$h'(a_1 \cdots a_k) = u_1 \cdots u_k = v_1 \cdots v_k = g'(a_1 \cdots a_k),$$

and hence $a_1 \cdots a_k = w'$ is a solution of I' and $h'(w') = w$.

For (iii), assume that $w' = a_1 \cdots a_k$ is a solution of I', that is, $h'(w') = g'(w')$. Now by case (i), $w = h'(w') = g'(w')$ is a solution of I.

Finally, case (iv) follows from the fact that $a \leq h(a)$ by assumption (4.3), and therefore if $\beta(a) = (u, v)$ exists, then $a \leq u$. □

Theorem 4.2. *Let $I = (h, g)$ be an instance of the marked PCP and $I' = (h', g')$ be its successor. Then I has a solution if and only if I' has a solution. Moreover, there is a one to one correspondence between the solutions of I and I'.*

Proof. Both claims follow obviously from the proofs of cases (ii) and (iii) of Lemma 4.4. □

4.2.2 Suffix complexity

We shall prove that the successor I' is simpler than the original instance I and then we use the construction of the successor inductively until we obtain an instance where the existence of a solution can be easily checked. We use two measures for the hardness of an instance. The first measure is the size of the domain alphabet. It is immediate that $|A'| \leq |A|$, since $A' \subseteq A$.

The second measure is the *suffix complexity*. For a morphism $h\colon A^* \to B^*$, we first define the *set of proper suffixes* by

$$S_h = \bigcup_{a \in A} \{x \mid x \prec h(a)\}.$$

The suffix complexity is defined to be the integer

$$\sigma(h) = |S_h|.$$

In other words, the suffix complexity is the number of proper suffixes of the images of the morphism. Note that by our definition ε is not a proper suffix. For an instance $I = (h, g)$ of the marked PCP, the suffix complexity is defined as

$$\sigma(I) = \sigma(h) + \sigma(g).$$

The next lemma was proved by Halava, Hirvensalo and de Wolf [21].

Lemma 4.5. *Let $I = (h, g)$ be an instance of the marked PCP and $I' = (h', g')$ be its successor. Then $\sigma(I') \leq \sigma(I)$.*

Proof. We define an injective function $p\colon S_{g'} \to S_h$. Let $w \in S_{g'}$, say $w \prec g'(a) = v = v_1 w$, where $\beta(a) = (u, v)$. Consider the factorization $u = u_1 b u_2$ such that $h(u_1) \leq g(v_1) < h(u_1 b)$, and let $s = g(v_1)^{-1} h(u_1 b)$ (see Figure 4.2). Clearly, $s \prec h(b)$ and $s \leq g(w)$. Note that w may be a suffix of many different images $g'(a)$, and therefore there can be several such suffixes s as defined above. Let s_1 be the shortest of these suffixes s. It is unique, since for all suffixes s in the above, $s \leq g(w)$. Define $p(w) = s_1$.

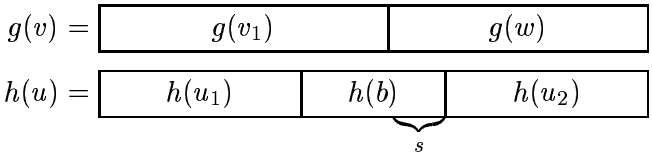

Figure 4.2: The suffix s corresponding to w

We show that p is injective. Suppose $w, w' \in S_{g'}$ and $p(w) = p(w') = s$. By the definition of s, there are words u and v such that

$$g(w) = sh(u) \quad \text{and} \quad g(w') = sh(v).$$

Since the blocks are minimal solutions and w and w' are suffixes in the corresponding blocks, we achieve that (w, u) and (w', v) are minimal solutions to the equation $sh(x) = g(y)$. Now it follows by Corollary 4.1 that necessarily $w = w'$ and $u = v$. Thus, p is injective, which implies $\sigma(g') \leq \sigma(h)$.

Similarly, we can define an injective function from $S_{h'}$ to S_g which proves that $\sigma(h') \leq \sigma(g)$. It now follows that $\sigma(I') = \sigma(h') + \sigma(g') \leq \sigma(g) + \sigma(h)$. □

Actually, the suffix complexity was already used in [5], where is was proved that Lemma 4.5 holds in the binary case. Here we proved that the suffix complexity in a successor sequence of any instance of the marked PCP cannot increase. This is a crucial part of our proof for the decidability of the marked PCP.

4.2.3 The marked PCP is decidable

Our decision procedure for the marked PCP uses the successors iteratively, i.e., it generates the successors of instances as long as we obtain an instance where the decision is easy to make. Therefore, we define the *successor sequence* as follows: let $I_0 = (h, g)$ be an instance of the marked PCP, where $h, g\colon (A_0)^* \to B^*$. We define the successor sequence $I_i = (h_i, g_i)$, $i \geq 0$, by $I_{i+1} = I_i'$. Moreover, let $A_i = A_{i-1}'$ for all $i \geq 1$, and thus, for all $i \geq 1$, $h_i, g_i\colon A_i^* \to A_{i-1}^*$. Recall that $A_0 \subseteq B$ and therefore the domain alphabets A_i satisfy $A_i \subseteq B$ for all $i \geq 0$.

We begin with a simple lemma.

Lemma 4.6. *Let A and B be alphabets with $A \subseteq B$ and let z be a positive integer. There exist only finitely many distinct instances $I = (h, g)$, where $h, g : A^* \to B^*$, of the PCP that satisfy $\sigma(I) \leq z$.*

Proof. Let $A = \{a_1, \ldots, a_m\} \subseteq B$ and assume that $|B| \geq 2$ and $m \geq 1$. An instance $I = (h, g)$ is completely specified by giving the $2m$ words $g(a_1), \ldots, g(a_m)$, $h(a_1), \ldots, h(a_m) \in B^*$. If one of those words has a length $> z+1$, then this word has more than z proper suffixes, and hence $\sigma(I) > z$. Accordingly, each of these $2m$ images can have a length of at most $z + 1$, and therefore there are $\sum_{i=1}^{z+1} |B|^i \leq |B|^{z+2}$ possible images for $|B| \geq 2$. We obtain that there are at most $|B|^{(z+2)2m}$ different possibilities for the $2m$ words. Hence, there are only finitely many different I that satisfy $\sigma(I) \leq z$. □

Now since the size of the alphabet and the suffix complexity do not increase, one of the following three cases occurs in the successor sequence I_i of an instance by Lemma 4.6:

(i) $|A_j| = 1$ for some $j \geq 0$,

(ii) $\sigma(I_j) = 0$ for some $j \geq 0$,

(iii) the sequence starts to cycle, i.e., there exist n_0 and $d \geq 1$ such that for all $j \geq n_0$, $I_j = I_{j+d}$, that is, the sequence is *ultimately periodic.*

Lemma 4.7. *In the instances that satisfy the condition (i) or the condition (ii), the existence of a solution can be determined.*

Proof. In case (i), we have an instance of the unary PCP. Since there is only one letter in the domain alphabet, the possible solutions are powers of this letter, and it is easy to see that if there is a solution, then also the letter itself is a solution.

In case (ii) we have suffix complexity zero, i.e., all the images are of length one, and therefore if there is a solution for this instance, then the first letter of such a solution is necessarily a solution. □

According to Lemma 4.7, to prove that the marked PCP is decidable, we need to prove that case (iii), the case of a *cycling sequence*, is also decidable.

Consider now a cycling case. In order to simplify the notations, we assume that the cycling starts already at I_0, i.e., the sequence is of the form

$$I_0, \ldots, I_{d-1}, I_d = I_0, \ldots \ . \tag{4.4}$$

This assumption can be made, since the original instance and the first instance in this sequence are equivalent and, by Lemma 4.6, there eventually is a cycle in every successor sequence. Note that the cycle can be easily detected.

Lemma 4.8. *Case (iii), a cycling sequence, is decidable. Moreover, in this case* (4.4) *the solution of minimal length is of length one.*

Proof. Let I_0 be as in (4.4) and $I_i = (h_i, g_i)$ for $i \geq 0$. By Lemma 4.4, for every solution x_i to some I_i, there is a solution x_{i+1} to I_{i+1} such that $x_i = h_{i+1}(x_{i+1}) = g_{i+1}(x_{i+1})$. Suppose x_0 is a solution to I_0 of minimal length. Inductively we obtain that there is a solution x_d to I_d such that

$$\begin{array}{ccccccccc} x_0 & = & h_1(x_1) & = & h_1h_2(x_2) & = & \ldots & = & h_1h_2\cdots h_d(x_d), \\ x_0 & = & g_1(x_1) & = & g_1g_2(x_2) & = & \ldots & = & g_1g_2\cdots g_d(x_d). \end{array}$$

Since the morphisms h_i and g_i cannot be length-decreasing, we have $|x_0| \geq |x_d|$. But since x_0 was chosen to be a minimal length solution to I_0 and x_d is

also a solution to $I_d = I_0$, $|x_0| = |x_d|$. This implies that $g_0 = g_d$ and $h_0 = h_d$ map the letters in x_d to letters. But then the first letter of x_d is already a solution, and hence $|x_0| = |x_d| = 1$. Thus I_0 has a solution if and only if I_0 has a 1-letter solution. □

We have proved the following theorem.

Theorem 4.3. *The marked PCP is decidable.*

Remark. It follows from our construction that it is decidable for an instance I of the marked PCP and for a given letter a whether or not I has a solution beginning with a.

4.2.4 The algorithm

We shall now summarize the whole algorithm for the marked PCP.

Algorithm 1. (*Decision procedure for the marked PCP*)

Let an instance $I = (h, g)$ be given.

(1) Set $c = 0$, $i = 0$, $I_0 = I$.

(2) Set $i = i + 1$.

(3) Construct $I_i = I'_{i-1}$.

(4) If I_i has a domain alphabet of size 1 or $\sigma(I_i) = 0$,
then decide I_i,
print the answer and terminate.

(5) If $|A_i| < |A_{i-1}|$ or $\sigma(I_i) < \sigma(I_{i-1})$
then set $c = i$ and goto 2.

(6) If $|A_i| = |A_{i-1}|$ and $\sigma(I_i) = \sigma(I_{i-1})$,
then check whether $I_i = I_j$ for some $c \leq j \leq i - 1$.
If there is such a j,
then check whether there is a 1-letter solution,
print the answer and terminate.
Else goto 2.

Let us consider the time complexity of the algorithm. Denote by N the size of an input, that is, the number of bits needed to give an instance as an input. Of course, the size of an instance of the PCP is the length of the sequence, where all the images are written one after another, separated by a special symbol.

First we consider the complexity of producing the blocks, i.e., finding the minimal length solutions to the equations $h(x) = h(y)$, where $a \leq h(x), g(y)$.

We define a simple procedure for the construction of a block $\beta(a)$ for a given $a \in B$. In the procedure the possible block is constructed letter by letter by defining a sequence (x_0, x_1) such that $h(x_0)$ and $g(x_1)$ are comparable at each step.

Algorithm 2. (Construction for $\beta(a)$).

Let (h, g) and a be given. Denote $h = h_0$ and $g = h_1$.

(1) Set $x_0 := a$, $x_1 := \varepsilon$, $s := h_0(x_0)$, $j := 1$, $S_1 = \{s\}$ and $S_0 = \emptyset$.

(2) If there exists $b \in A$ such that s and $h_j(b)$ are comparable
then goto (3); else return $\beta(a) = \emptyset$.

(3) Set $x_j := x_j b$ and

(3.1) if $h_j(b) \leq s$ then $s := h_j(b)^{-1} s$;

(3.2) if $s \leq h_j(b)$ then $s := s^{-1} h_j(b)$ and $j := 1 - j$.

(4) If $s = \varepsilon$ then return $\beta(a) := (x_0, x_1)$;
else if $s \in S_j$ then return $\beta(a) = \emptyset$;
else set $S_j := S_j \cup \{s\}$ and goto (2).

The suffix s in the above algorithm is called an *overflow* (*of h or g*, if $j = 0$ or $j = 1$, respectively), and the set S_0 is the *set of overflows of h* and S_1 is the *set of overflows of g*. Note that there are three possible ways for the algorithm to terminate:

(i) There is no suitable b such that $h_j(b)$ and s are comparable.

(ii) The same overflow s of h (resp. g) occurs twice.

(iii) The overflow equals $s = \varepsilon$.

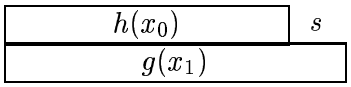

Figure 4.3: Overflow s of h.

First of all, the sequence (x_0, x_1) is constructed letter by letter and there are only finitely many different possible overflows, since $s \in S_h \cup S_g$. This implies that the algorithm eventually terminates in case (ii) if it does not meet the other two cases. In fact, since the construction of the sequence (x_0, x_1) is deterministic, case (ii) implies that the overflows form an ultimately periodic sequence, and we would never obtain cases (i) or (iii). In this case we know that $h(x_0)$ and $g(x_1)$ are comparable forever, and the overflows are always nonempty. Clearly, there cannot exists any solution to the equation $h(x) = g(y)$ beginning with a in this case.

It is clear that $\beta(a)$ exists if and only if the algorithm terminates in case (iii). We note that although case (ii) is not important in the marked PCP, it needs to be studied when we consider the infinite solutions of the marked PCP in Section 4.5.

Algorithm 2 for constructing a block is deterministic, because g and h are marked. Furthermore, if N is the length of the instance I given as input, then the algorithm runs in time polynomial in N. Namely, each $g(a_i)$ and $h(a_i)$ can have a length at most N, and hence they can have at most $N-1$ proper suffixes. Since there are $2n = O(N)$ different images $g(a_i)$ and $h(a_i)$, where n is the size of the instance, there are only $O(N^2)$ different suffixes, and hence the search for the new letter b is done at most $O(N^2)$ times. For a particular b the checking of whether $h_j(x_j)$ and s are comparable can be done in $O(N)$ steps. Therefore, Algorithm 2 runs in $O(N^3)$ steps. Now, since there are $O(N)$ letters for which the block is constructed in step (3) of Algorithm 1, we conclude that step (3) can be completed in $O(N^4)$ steps.

Step (4) of Algorithm 1, on the other hand, where we compute the size of the alphabet, suffix complexity and finally the possible one letter solutions, can be trivially done in $O(N^2)$ time.

Consider now the worst case complexity of Algorithm 1 for the marked PCP for a fixed size of the alphabet and suffix complexity z. By the proof of Lemma 4.6, there are at most $|B|^{(z+2)2n}$ successors to be generated and, since $|B| = O(N)$ and $z = O(N^2)$, we know that there are

$$O(N^{(N^2+2)2N}) = O(N^{N^3}) = O(2^{N^3 \cdot \log N}), \tag{4.5}$$

different successors to be generated, which is the number of times we repeat step (3). To construct a new successor, and to check whether this instance has already occurred in the sequence requires $2^{O(\log N \cdot N^3)}$ time. Indeed, the equality of two instances can be checked in linear time, the construction for the next successor was done in time $O(N^4)$, and we have at most $2^{O(\log N \cdot N^3)}$ instances to be compared. Therefore, the complexity of Algorithm 1 for a fixed size of the alphabet and suffix complexity is $2^{O(\log N \cdot N^3)}$.

In the successor sequence there are $n = O(N)$ possible sizes of the alphabets and $z = O(N)$ possible suffix complexities to go through, and therefore

the complexity of the algorithm is

$$O(N) \cdot O(N) \cdot 2^{O(\log N \cdot N^3)} = O(2^{N^3 \cdot \log N}).$$

On the other hand, since the proof of Lemma 4.6 actually gives us a computable upper-bound for the number of different instances, we could also use a counter to be sure that we have a loop without making any comparisons between the instances. Indeed, we could construct a counter to count the number of successors constructed in the sequence with a constant size of the alphabet and suffix complexity. If one of these measures decreases, then the counter is reset to zero. Once the counter breaks the bound of Lemma 4.6 for some fixed size of the alphabet and suffix complexity, the sequence is necessarily cyclic. Then we simply check whether or not the present instance has a one letter solution.

For a number m, the length of the binary representation of m is $O(\log m)$. Therefore, by (4.5), the counter needs space for

$$\log(O(2^{N^3 \cdot \log N})) = O(N^4)$$

bits. By using the counter method, we obtain an algorithm with polynomial space complexity.

We have thus

Theorem 4.4. *The marked PCP is in* **EXPTIME** *and* **PSPACE**.

Remark. It is known that $PSPACE \subseteq EXPTIME$, but whether the inclusion is proper is not known.

Example 4.2. Let us illustrate the algorithm for the marked PCP with the following simple example. This example gives us many details about the behaviour of the algorithm and the successor sequence.
Let $I_0 = (h_0, g_0)$ be an instance of the marked PCP where $h_0, g_0 \colon \{a,b,c\}^* \to \{a,b,c\}^*$

	a	b	c
h_0	abb	bb	$cbbb$
g_0	a	bbb	c

The sets of proper suffixes of the instance are: $S_{h_0} = \{b, bb, bbb\}$ and $S_{g_0} = \{b, bb\}$. Therefore, the suffix complexity $\sigma(I_0) = 5$. The blocks for the letters are

$$\beta(a) = (abb, abb), \quad \beta(b) = (bbb, bb) \quad \text{and} \quad \beta(c) = (c, cb).$$

The successor instance I_1 is now

	a	b	c
h_1	abb	bbb	c
g_1	abb	bb	cb

The instance I_1 has the same size as I_0, but the suffix complexity has decreased to 4, since $S_{h_1} = \{b, bb\}$ and $S_{g_1} = \{b, bb\}$. Now the blocks of I_1 are

$$\beta(a) = (a, a), \quad \beta(b) = (bb, bbb) \quad \text{and} \quad \beta(c) = (cb, cb),$$

and therefore instance I_2 is as follows:

	a	b	c
h_2	a	bb	cb
g_2	a	bbb	cb

Again, the size of I_2 is equal to the size of I_0, and the suffix complexity decreases, since $S_{h_2} = \{b\}$ and $S_{g_2} = \{b, bb\}$, and $\sigma(I_2) = 3$.

The blocks of I_2 are

$$\beta(a) = (a, a), \quad \beta(b) = (bbb, bb) \quad \text{and} \quad \beta(c) = (c, c),$$

and we obtain that I_3 is

	a	b	c
h_3	a	bbb	c
g_3	a	bb	c

The suffix complexity is still 3, since $S_{h_3} = \{b, bb\}$ and $S_{g_2} = \{b\}$. The blocks are

$$\beta(a) = (a, a), \quad \beta(b) = (bb, bbb) \quad \text{and} \quad \beta(c) = (c, c),$$

and I_4 is

	a	b	c
h_4	a	bb	c
g_4	a	bbb	c

Now the blocks are

$$\beta(a) = (a, a), \quad \beta(b) = (bbb, bb) \quad \text{and} \quad \beta(c) = (c, c),$$

and clearly $I_5 = I_3$. Note that I_3 is the first instance of the cycle.

Let us now point out a few things about the successor sequence in this example. First, from I_3 we deduce that there exist solutions beginning with a and c. Actually, the existence of a solution for a was already seen in I_2, even though the cycle begins from I_3 (this is where the solution for c is found). Therefore, not all solutions are found at the same instance of the successor sequence. Clearly, the length of a solution effects the number of reductions needed to find this solution. Indeed, let us look at the minimal solutions for a and c. First of all, for a the minimal solution is abb and $\beta(a) = (abb, abb)$, i.e., the block decomposition consists of a single block. But the minimal solution for c is $cbbb$, where $\beta(c) = (c, cb)$ and $\beta(b) = (bbb, bb)$, i.e., the solution has the block decomposition of length two.

The second remark is about the suffix complexity. Note that although the suffix complexities are equal in I_2 and I_3, the cycle begins from I_3, i.e., the cycle does not begin from the first instance with suffix complexity three.

Therefore, it is not immediately clear from which instance the cycle in the sequence begins.

Remark. The algorithm for the marked PCP has been implemented in the binary case by students M. Laine, M. Rosten and M. Varjonen. The examples were considered in the marked instances over a binary alphabet and they indicated that the binary case is quite simple.

4.3 The 2-Marked PCP is undecidable

Recall that a morphism h is k–marked if each image has a unique prefix of length k, i.e., for all letters a and b

$$\mathrm{pref}_k(h(a)) \neq \mathrm{pref}_k(h(b)),$$

We shall next prove that the 2–marked PCP is undecidable.

Note that the fact that an instance is 2-marked does not imply injectivity, since, for example, h, where $h(a) = a$, $h(b) = ab$, $h(c) = b$, is 2–marked, but $h(b) = ab = h(ac)$. The decidability status of the PCP is open for the *strongly* 2–marked morphisms, where the morphisms are 2–marked and no image is a prefix of another image.

4.3.1 Tzeitin semigroup

The proof of the undecidability of the 2–marked PCP uses the undecidability of the word problem in semigroup S_7 defined by Tzeitin [38]. Consider the following semigroup S_7 with a set of five generators $\Gamma = \{a, b, c, d, e\}$ and seven relations:

$$S_7 = \langle a, b, c, d, e \mid R \rangle,$$
$$R = \{ac = ca, ad = da, bc = cb, bd = db, eca = ce, edb = de, cca = ccae\}$$

Here the relations in R are the 'rewriting rules' of the semigroup.

Tzeitin [38] proved the following theorem.

Theorem 4.5. *It is undecidable, for words $u, v \in \Gamma^*$, whether or not $u = v$ in S_7.*

The problem of whether $u = v$ in a semigroup S is called the *word problem of S*. Recall that $u = v$ in S if and only if there is a sequence u_i, $i = 0, 1, \ldots, n$, such that $u_0 = u$, $u_n = v$ and for $i = 0, 1, \ldots n-1$,

$$u_i = x_i\alpha_i y_i, \quad u_{i+1} = x_i\beta_i y_i \text{ and } (\alpha_i = \beta_i) \in R,$$

where R is the set of relations of S. This is denoted by $u = v \in S$.

Note that in the set of relations R of S_7, the set of the seven left hand sides of R is 2-marked, and similarly for the set of the seven right hand sides of R. We will reduce the 2-marked PCP to the word problem of S_7. We use a slight modification of the standard reduction from word problems to the PCP, involving an alphabet with underlined letters in order to ensure that the morphisms are 2-marked.

Define the domain alphabet as

$$A = \Gamma \cup \underline{\Gamma} \cup \{F, E, \#, \underline{\#}, r_1, r_2, \ldots, r_7, \underline{r_1}, \underline{r_2}, \ldots, \underline{r_7}\},$$

where $\underline{\Gamma} = \{\underline{a}, \underline{b}, \underline{c}, \underline{d}, \underline{e}\}$, and $r_1, \ldots, r_7$ are the 7 relations in R and $\underline{r_1}, \ldots, \underline{r_7}$ are their underlined versions (considered as single letters), so that $r_1 = [ac = ca]$, $\underline{r_1} = [\underline{ac} = \underline{ca}]$, etc. Define the range alphabet as

$$B = \Gamma \cup \underline{\Gamma} \cup \{F, E, \#, \underline{\#}\}.$$

The letters F and E will mark the beginning and the end of the expressions, respectively. For given $u, v \in \Gamma^+$, g and h are defined in Table 4.1.

	F	E	$\#$	$\underline{\#}$	x	$\underline{x}$	$[s = t]$	$[\underline{s} = \underline{t}]$
g	$Fu\#$	E	$\underline{\#}$	$\#$	$\underline{x}$	x	$\underline{t}$	s
h	F	$\underline{\#}vE$	$\#$	$\underline{\#}$	x	$\underline{x}$	s	$\underline{t}$

Table 4.1: Definitions of g and h, where $x \in \Gamma$ and $s = t \in R$

Note that the constructed instance $I = (h, g)$ is an instance of the 2-marked PCP. The following lemma shows that the word problem for u and v is equivalent to the instance I of the PCP.

Lemma 4.9. *Let u, v and I be as above. Then $u = v \in S_7$ if and only if I has a solution.*

Proof. First, suppose $u = v \in S_7$. Then there is a sequence $u = u_1, u_2, \ldots, u_k = v$, where $u_i = u_i' s u_i''$ and $u_{i+1} = u_i' t u_i''$, and $s = t \in R$. We construct a solution to I by induction on k.

If $k = 1$, then $u = v$ (in Γ^+). Now $x = Fu\#\underline{u}E$ is a solution to I.

Assume now that $k \geq 2$. Let $I_0 = (g_0, h_0)$, where $g_0, h_0 \colon A^* \to B^*$, be the instance of the 2-marked PCP corresponding to $u = u_{k-1} \in S_7$. By the induction hypothesis, I_0 has a minimal length solution x'. It is easy to see that every solution must begin with F and end with E, so that $x' = FyE$. Let $w = h_0(Fy)$. Then $g_0(Fy) = w\#u_{k-1}$. Note that since I and I_0 only differ in the assignments $h(E)$ and $h_0(E)$, and E cannot occur in y (because x' is minimal), we also have $g(Fy) = w\underline{\#}u_{k-1}$ and $h(Fy) = w$. We distinguish two cases. Firstly, let $u_{k-1} = u's u''$ and $v = u_k = u't u''$, where $r = [s = t] \in R$. Then it is easily verified that $x = Fy\underline{\#}u'ru''\#\underline{u't}u''E$

is a solution to I. Secondly, if $u_{k-1} = u'tu''$ and $v = u_k = u'su''$, then $x = Fy\underline{\#}u'tu''\#\underline{u'ru''}E$ is a solution. This completes the induction step.

For the other direction, suppose I has a solution x. We can assume that x is of minimal length. Clearly, this x must be of the form $Fx_1x_2\cdots x_mE$, where $x_i \in A$. Hence $g(x) = Fu\#g(x_1\cdots x_m)E = h(x) = Fh(x_1\cdots x_m)\underline{\#}vE$. Ignoring the underlining, $g(x) = h(x)$ must be of the form

$$Fu_1\#u_2\#\cdots\#u_{k-1}\#u_kE,$$

where $u_i \in \Gamma^*$, $u_1 = u$ and $u_k = v$. We show that $u_i = u_{i+1} \in S_7$ for every $1 \le i \le k-1$, from which $u = v \in S_7$ follows.

Because $\#$ occurs in $h(x_1\cdots x_m)$, there must be a least i such that $x_i = \#$, and hence $u = h(x_1\cdots x_{i-1})$. Since there is no underlining in u, it follows that $x_1, \ldots, x_{i-1}$ are chosen from $\Gamma \cup R$. Let $x_1\cdots x_{i-1} = w_1r_{i_1}w_2r_{i_2}\cdots w_l$, with $w_i \in \Gamma^*$ and $r_i = [s_i = t_i] \in R$. Then $u = h(w_1r_{i_1}w_2r_{i_2}\cdots w_l) = w_1s_{i_1}w_2s_{i_2}\cdots w_l$, see Figure 4.4 for an illustration.

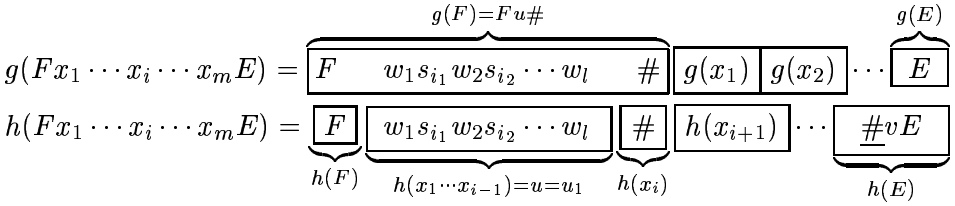

Figure 4.4: An illustration for the proof.

Note that $g(x_1\cdots x_{i-1}) = g(w_1r_{i_1}w_2r_{i_2}\cdots w_l) = \underline{w_1t_{i_1}w_2t_{i_2}\cdots w_l}$. But, since $g(x_1\cdots x_mE) = h(x_{i+1}\cdots x_mE)$, there must be the least $j > i$ such that $x_j \in \{\#, \underline{\#}\}$ and $h(x_{i+1}\cdots x_{j-1}) = g(x_1\cdots x_{i-1}) = \underline{w_1t_{i_1}w_2t_{i_2}\cdots w_l}$. The latter string (without underlining) is u_2. Note that $u_1 = u_2 \in S_7$, because $u_1(= u)$ and u_2 only differ by u_2 having t_i where u_1 has s_i.

Continuing this reasoning, we conclude that for every two words $u_i, u_{i+1} \in \Gamma^*$ occurring in $g(x) = h(x)$ and separated by $\#$, ignoring the underlining we have $u_i = u_{i+1} \in S_7$. (Some of the words u_i and u_{i+1} may actually already be equal in A^+.) Hence, $u = v \in S_7$, since $g(x)$ starts with $u_1 = u$ and ends with $u_k = v$. □

Together with Tzeitin's result, the above lemma implies:

Theorem 4.6. *The 2-marked PCP is undecidable.*

As mentioned, 2-marked morphisms are not special cases of the injective morphisms. The next theorem follows by the construction in Ruohonen [36], since the biprefix instances of the PCP constructed there are actually 5-marked.

Theorem 4.7. *The strongly 5-marked PCP is undecidable.*

Problem 4.1. *Is the strongly k-marked PCP decidable for $1 < k < 5$?*

4.4 Equality sets of the marked PCP

In this section we consider the set of all solutions of an instance of the marked PCP. Note that the algorithm for the marked PCP on page 35 not only decides the marked PCP, but also tells something more about the solutions.

Assume that $I = (h, g)$ is an instance of the marked PCP, where $h, g\colon A^* \to B^*$. As earlier, we can assume that $A \subseteq B$ and that $a \leq h(a)$ for all $a \in A$. In the successor sequence of I, by Lemma 4.4, for all $i \geq 0$, $a \leq h_i(a)$ if $h_i(a)$ exists.

Our algorithm can be transformed into such a form that it also outputs the letters for which there eventually exists a 1-letter solution. Then for each such letter a, there is a solution of I beginning with a, by the assumption made on h. Assume that the algorithm outputs a subset C of A with $a \in C$ if and only if there is a solution beginning with a for I. Of course, we can effectively find the solutions for these letters in C by simply generating them from the beginning. The next theorem is now immediate.

Theorem 4.8. *For an instance I of the marked PCP, the equality set is*

$$E(I) = \{w_1, w_2, \ldots, w_k\}^*$$

where the words w_i are the minimal solutions of the instance. Moreover, $E(I)$ is regular and the (generating) set $\{w_1, w_2, \ldots, w_k\}$ can be effectively found.

4.5 Infinite solutions

We shall end this chapter by considering infinite solutions of the marked PCP. We say that the morphisms h and g *agree* on an *infinite word* $\omega = a_1 a_2 \cdots$ if $g(w)$ and $h(w)$ are comparable for all $w \leq \omega$. We call such an infinite word an *infinite solution* of the instance (h, g). The following theorem is from Ruohonen [36].

Theorem 4.9. *It is undecidable whether there is an infinite solution to an instance (h, g) of the PCP.*

Actually, by [36], it can be assumed that the morphisms are biprefix morphisms.

We shall prove that the existence of an infinite solution is decidable for the marked PCP. Our construction for the marked PCP turns out to be very useful also in this case, and we shall use the notation and terminology introduced earlier in this chapter. The proof is from Halava and Harju [13].

We begin with a simple lemma.

Lemma 4.10. *Let $I = (h, g)$ be an instance of the marked PCP over an alphabet A and $I' = (h', g')$ be its successor over A'. For all words $x, y \in (A')^*$ such that x and y are comparable, $h(h'(x))$ and $g(g'(y))$ are also comparable.*

Proof. Assume, by symmetry, that $x \leq y$, say $y = xz$ for a word z. By Lemma 4.4, $h(h'(x)) = g(g'(x))$, and therefore $h(h'(x)) \leq g(g'(xz))$ as required. □

Assume that ω is an infinite solution of an instance $I = (h, g)$ of the marked PCP. We say that ω has a *block decomposition* if

$$\omega = u_1 u_2 \cdots = v_1 v_2 \cdots ,$$

where $(u_i, v_i) = \beta(a_i)$ for a letter a_i for each i. There are three possible cases for ω:

(i) $\omega = w_1 w_2 \cdots$, where $w_i \in E(I)$ for each i,

(ii) ω has a block decomposition, but it is not as in (i),

(iii) ω does not have a block decomposition.

Example 4.3. Let h and g be as in Example 4.2,

	a	b	c
h	abb	bb	$cbbb$
g	a	bbb	c

It was shown in Example 4.2 that $E(h, g) = \{abb, cbbb\}^*$. Now, for example, $\omega_1 = abbcbbbabbcbbb\cdots$ is an infinite solution of the type (i) of the instance (h, g). On the other hand

$$\omega_2 = bb\cdots$$

is an infinite solution with a block decomposition (type (ii)), since $\beta(b) = (bb, bbb)$.

For type (iii), let $h(e) = ed$, $h(d) = dd$, $g(e) = e$ and $g(d) = dd$. Now $\omega_3 = eddd\cdots$ is an infinite solution without a block decomposition, since $\beta(e)$ does not exist.

First of all, the solutions of type (i) can be effectively found by Theorem 4.8. Note that if there exists a solution to the marked PCP, then there exists an infinite solution. Therefore, we assume in the following that $E(I) = \{\varepsilon\}$ and consider only solutions of the other two types.

Again, cases where the suffix complexity is zero or the domain alphabet is unary are easy to decide.

Lemma 4.11. *Let I be an instance of the marked PCP. If $\sigma(I) = 0$ or the domain alphabet is unary, then the infinite solutions of I can be effectively found.*

Proof. Assume first that $\sigma(I) = 0$. Then for all letters, the lengths of the images are one. It is obvious that I has an infinite solution if and only if it has a finite solution.

Assume that I is unary and let a be this single letter. Then $I = (h, g)$ has an infinite solution if and only if $h(a)^k = g(a)^\ell$ for some k and ℓ. This follows from the fact that $h(a)^t$ and $g(a)^t$ have to be comparable for all $t \geq 0$, and therefore, for some k and ℓ, $k \cdot |h(a)| = \ell \cdot |g(a)|$. If $h(a)^k = g(a)^\ell$, then $\omega = aa\cdots$ is an infinite solution of I. □

We shall prove that a solution of the type (ii) of an instance I reduces to an infinite solution in the successor instance I'.

Lemma 4.12. *Let $I = (h, g)$ be an instance of the marked PCP with $E(I) = \{\varepsilon\}$. There is an infinite solution ω with a block decomposition if and only if the successor $I' = (h', g')$ has an infinite solution. Moreover, these solutions begin with the same letter.*

Proof. Assume first that there exists an infinite solution ω of type (ii) of I. Then ω has two factorizations,

$$\omega = u_1 u_2 \cdots = v_1 v_2 \cdots , \tag{4.6}$$

where $(u_i, v_i) = \beta(a_i)$ for a letter a_i. Clearly, $h(a_1)$ and $g(a_1)$ are comparable and $h'(a_1) = u_1$ and $g'(a_1) = v_1$ are comparable. By the assumption on h, they both begin with a_1. Define now $\omega' = a_1 a_2 \cdots$. By (4.6) it is obvious that ω' is an infinite solution of I'.

Similarly, if the successor $I' = (h', g')$ has an infinite solution, say $\omega' = a_1 a_2 \cdots$, then, by the definition, for each i, $\beta(a_i) = (u_i, v_i)$ for some words u_i and v_i. Clearly, $\omega = u_1 u_2 \cdots = v_1 v_2 \cdots$ is an infinite solution of type (ii). □

By Lemma 4.12, for the solutions of type (ii), instead of searching for an infinite solution of I, we can turn to the simpler instance I'. The difficulty here is that we do not know whether a possible solution of the successor is of type (ii) or (iii).

We shall first prove a case where for the entire successor sequence an infinite solution is of type (ii), i.e., each I_i has a block decomposition in all instances of the successor sequence.

Consider now the successor sequence I_i, where $I = I_0$ and $I_i = (h_i, g_i)$. Assume that n_0 and $d \geq 1$ are such that, for all $j \geq n_0$, $I_j = I_{j+d}$.

Lemma 4.13. *Let I_i, $i = 0, 1, \ldots$, be the successor sequence for an instance $I_0 = (h_0, g_0)$ of the marked PCP with $E(I_0) = \{\varepsilon\}$, where $h_i, g_i \colon A_i^* \to A_{i-1}^*$. There exists an infinite solution of type (ii) for all I_i if and only if there exists $b \in A_0$ such that $h_i(b)$ and $g_i(b)$ are comparable for all $i \geq 0$.*

Proof. Assume first that there is an infinite solution ω_i of I_i for each i and assume that ω_i is obtained from ω_{i-1} as in the first part of the proof of Lemma 4.12. Let b be the first letter of ω_0. Then, by the assumption $b \leq h_0(b)$, we obtain that $b \leq g_0(b)$ and $h_0(b)$ and $g_0(b)$ are comparable. Furthermore, there is a block for the letter b, and therefore, by the construction in the proof of Lemma 4.12, $b \leq \omega_1$. Inductively, we find that b is the first letter of ω_i for all $i \geq 0$. Therefore, there is a block in each I_i, and $h_i(b)$ and $g_i(b)$ are comparable for all i.

In the other direction, assume that there exists a letter b for which $h_i(b)$ and $g_i(b)$ are comparable for all i. We shall first prove that there is an infinite solution of type (ii) for I_0. By Lemma 4.10, the words

$$x_i = h_1(\cdots h_{i-1}(h_i(b))\cdots) \quad \text{and} \quad y_i = g_1(\cdots g_{i-1}(g_i(b))\cdots)$$

are comparable and they begin with b for all i by the assumption. Let $z_i = x_i \wedge y_i$. Clearly, z_i is either x_i or y_i, depending on the case. Since the morphisms in the successor sequence are nonerasing and h_1 and g_1 are marked, we obtain that $z_i \leq z_{i+1}$. Now the word $\omega = \lim_{i\to\infty} z_i$ is an infinite solution to I_0, since, by Lemma 4.10, $h_0(h_1(\cdots h_{i-1}(h_i(b))\cdots))$ and $g_0(g_1(\cdots g_{i-1}(g_i(b))\cdots))$ are comparable. The claim follows inductively by replacing I_0 by I_i. □

We have considered the two simple cases of our problem. These two are the cases where an infinite solution can be detected simply by the algorithm for the marked PCP. We shall next prove that the infinite solutions of type (iii) can also be detected.

Lemma 4.14. *It is decidable whether an instance $I = (h, g)$ of the marked PCP with $E(I) = \{\varepsilon\}$ has an infinite solution without a block decomposition.*

Proof. Let $I = I_0$ and I_i, $i = 1, 2, \ldots$, be the successor sequence of I, where $I_i = (h_i, g_i)$, where $h_i, g_i \colon A_i^* \to A_{i-1}^*$. Assume that ω is an infinite solution of I without a block decomposition, that is,

$$\omega = u_1 u_2 \cdots u_n \omega_1 = v_1 v_2 \cdots v_n \omega_2, \tag{4.7}$$

where $(u_i, v_i) = \beta(a_i)$ for some letters a_i for $1 \leq i \leq n$, and ω_1 and ω_2 are infinite words which do not have a block as a prefix. Note that also $n = 0$ is possible.

We shall first prove that $a_1 \notin A_i$ for some i, i.e., a_1 *disappears* in the successor sequence. Assume to the contrary that there is a block for a_1 in all instances I_i. Now $h(a_1)$ and $g(a_1)$ are necessarily comparable and $a_1 \leq h(a_1) \wedge g(a_1)$. Therefore for all instances, $a_1 \leq h_i(a_1) \wedge g_i(a_1)$. In order to have a block for a_1 in every instance I_i, necessarily $h_i(a_1)$ and $g_i(a_1)$ are comparable for all $i \geq 0$. By Lemma 4.13, there is an infinite solution for I with a block decomposition beginning with a_1. Moreover, an

infinite solution beginning a_1 is unique by the fact that the morphisms are marked and $E(I) = \{\varepsilon\}$, which is a contradiction, since ω has no block decomposition.

The disappearing letters can be effectively found when constructing the successor sequence.

Also, the prefixes $u_1 \cdots u_n$ and $v_1 \cdots v_n$ in (4.7) can be effectively found. Indeed, we construct a sequence (x_i, y_i) of blocks of the instance I as follows: $(x_1, y_1) = (u_1, v_1)$, where $\beta(a_1) = (u_1, v_1)$, and $(x_{i+1}, y_{i+1}) = (x_i u_{i+1}, y_i v_{i+1})$, where (u_{i+1}, v_{i+1}) is the unique block satisfying the conditions

- $h(x_i u_{i+1})$ and $g(y_i v_{i+1})$ are comparable, and
- $x_i u_{i+1}$ and $y_i v_{i+1}$ are comparable.

Since a_1 disappears, by Lemma 4.13, there cannot be an infinite solution with a block decomposition beginning with a_1, and hence this process ends. Denote by (x, y) the final pair achieved by this process.

Next we shall consider the words ω_1 and ω_2. It is clear that $h(\omega_1) = g(\omega_2)$, since $h(u_i) = g(v_i)$ for all $i = 1, 2, \ldots, n$. Let $b \leq h(\omega_1)$. Thus, b is a letter for which there is no block in I, which means that

$$\omega_1 = uu'u' \cdots \quad \text{and} \quad \omega_2 = vv'v'v' \cdots \tag{4.8}$$

for some words u, u', v and v' with $|h(u')| = |g(v')|$. There are only finitely many letters b such that there does not exist a block for b in I, and therefore all possible pairs ω_1 and ω_2 can be effectively expressed.

Let (x, y) be as defined in the above. Then, $h(x\omega_1) = h(x)h(\omega_1) = g(y)g(\omega_2) = g(y\omega_2)$, and therefore this is an infinite solution only if $x\omega_1 = y\omega_2$. Let $k = \max\{|xu|, |yv|\}$. We check first whether the prefixes of length k of $x\omega_1$ and $y\omega_2$ are equal. If not then $x\omega_1$ is not a solution. Otherwise, let x' be the common prefix of length k of $x\omega_1$ and $y\omega_2$. Now

$$x\omega_1 = x'z_1z_2z_1z_2z_1 \cdots \text{ and } y\omega_2 = x'r_1r_2r_1r_2r_1 \cdots ,$$

where $z_2z_1 = u'$ and $r_2r_1 = v'$ (actually, either z_1 or r_1 is ε). Hence, $x\omega_1 = y\omega_2$ if and only if $(z_1z_2)^t = (r_1r_2)^\ell$ for some t and ℓ. Such (minimal) integers t and ℓ can be effectively found if they exist, since we should have

$$|z_1z_2| \cdot t = |r_1r_2| \cdot \ell.$$

Since the pair (x, y) is unique for a_1 and there are only finitely many possible ω_1 and ω_2, we can check all possible words $x\omega_1$ and $y\omega_2$ to find out whether they are infinite solutions of I. Finally, note that we also have to check whether the words ω_1 and ω_2 are solutions themselves. This is the case where $n = 0$. □

Corollary 4.2. *It is decidable whether an instance of the marked PCP has an infinite solution of type (ii).*

Proof. We have proved that it is decidable whether there is a solution that reduces to an infinite solution with a block decomposition for all successors in the sequence. If this is not the case, then an infinite solution reduces to an infinite solution without a block decomposition, by Lemma 4.13, in some reduction step of the successor sequence. Therefore, by checking the infinite solutions without a block decomposition for all (finitely many) instances in the successor sequence we can also detect these solutions. By Lemma 4.12, these can be transformed to infinite solutions of the original instance. □

We have proved the following theorem.

Theorem 4.10. *It is decidable whether an instance of the marked PCP has an infinite solution.*

Chapter 5

The marked generalized Post Correspondence Problem

In this chapter we shall extend the decidability result of the marked Post Correspondence Problem to the marked generalized Post Correspondence Problem (GPCP). Using this result we also prove that the binary Post Correspondence Problem is decidable.

As in Chapter 3, we denote an instance of the GPCP by

$$I = ((p_1, p_2), h, g, (s_1, s_2)),$$

where h, g: $A^* \to B^*$ are morphisms and $p_1, p_2, s_1, s_2 \in B^*$. Recall that the GPCP is to determine whether or not there exists a solution $w \in A^*$,

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

In this chapter we study the marked GPCP, i.e., we assume that h and g are marked. We shall call the pair (s_1, s_2) in the instance an *end pair*.

As mentioned earlier, Ehrenfeucht, Karhumäki and Rozenberg in [5], when proving the decidability of the binary PCP, actually proved that the binary marked GPCP is decidable. Then it was shown that every binary instance of the PCP is either periodic or can be reduced to an instance of the binary GPCP with marked morphisms, and the decidability of the binary PCP followed. We shall present this technique in Section 5.4.

Our main result of this chapter is the following.

Theorem 5.1. *The GPCP with marked morphisms is decidable for the alphabets of any size.*

Therefore the new proof for the binary PCP is achieved in this chapter. We shall consider the details of the binary case in the last section of this chapter.

5.1 Successors

Our proof of Theorem 5.1 uses the same idea as the proof for the decidability of the marked PCP, Theorem 4.3. We shall again use the concept of a successor, but in the marked GPCP the begin and the end words make the process a bit different. Actually, in the marked GPCP, an instance does not have a unique successor, but it can have infinitely many successors. Nevertheless, we are capable of choosing the right successors in order to determine the possible solutions.

5.1.1 Modified instances

Let $I = ((p_1, p_2), h, g, (s_1, s_2))$, where $h, g\colon A^* \to B^*$, be an instance of the marked GPCP. Here we again assume that

$$A \subseteq B \text{ and } a \leq h(a) \text{ for all } a \in A. \tag{5.1}$$

If w is a solution of I, then it is called *minimal* if, for all $v \neq \varepsilon$ such that $w = w_0 v w_1$, $w_0 w_1$ is not a solution of I.

The instances

$$I = ((p_1, p_2), h, g, (s_1, s_2)) \tag{5.2}$$

can be reduced to instances where $p_1 = \varepsilon$ or $p_2 = \varepsilon$ and $s_1 = \varepsilon$ or $s_2 = \varepsilon$, since to have a solution, necessarily

$$(p_1 \leq p_2 \text{ or } p_2 \leq p_1) \text{ and } (s_1 \preccurlyeq s_2 \text{ or } s_2 \preccurlyeq s_1). \tag{5.3}$$

For instance, if $p_1 \leq p_2$ and $s_2 \preccurlyeq s_1$, say $p_2 = p_1 p$ and $s_1 = s s_2$, then (5.2) is equivalent to $((\varepsilon, p), h, g, (s, \varepsilon))$. If the words p_i and s_i do not fulfill the condition (5.3), then the instance is called *trivial*.

We first modify the marked GPCP by requiring that the solutions begin with a new fixed letter #. For this, let # be a new letter, i.e., $\# \notin B$. If in (5.2) $p_1 \neq \varepsilon$ or $p_2 \neq \varepsilon$, we extend the morphisms by defining $h(\#) = \#p_1$ and $g(\#) = \#p_2$. On the other hand, if $p_1 = \varepsilon = p_2$, we fix a letter $a_0 \in A$ and define $h(\#) = \#h(a_0)$ and $g(\#) = \#g(a_0)$.

In both cases, the extended morphisms $h, g\colon (A \cup \{\#\})^* \to (B \cup \{\#\})^*$ remain marked, and in the latter case, (5.2) has a solution that begins with a_0 if and only if the modified instance $((\varepsilon, \varepsilon), h, g, (s_1, s_2))$ has a solution in $\#A^*$. Clearly, the marked GPCP is decidable if we can decide whether a solution exists for some $a_0 \in A$.

To simplify the notations, we shall assume that $\# \in A$ and $\# \in B$. Therefore, we can confine ourselves to the instances of the form

$$(\#, h, g, (s_1, s_2)) \qquad (s_1 = \varepsilon \text{ or } s_2 = \varepsilon), \tag{5.4}$$

where the solutions w are required to satisfy $w \in \#(A \setminus \{\#\})^*$ together with $h(w)s_1 = g(w)s_2$. We shall refer this modified GPCP in (5.4) simply by GPCP.

If an instance (5.4) has $s_1 = s_2 = \varepsilon$, then we have to check whether there is a solution of the marked PCP beginning with the letter #, but this is decidable by Theorem 4.3. Therefore, we shall assume that $s_1 s_2 \neq \varepsilon$. Hence, we state

Lemma 5.1. *The GPCP for marked morphisms is decidable if and only if it decidable for the instances* $(\#, h, g, (s_1, s_2))$, *where* $s_1 = \varepsilon$ *or* $s_2 = \varepsilon$, $s_1 s_2 \neq \varepsilon$.

5.1.2 Blocks and successors of instance

Let $I = (\#, h, g, (s_1, s_2))$, where $h, g\colon A^* \to B^*$, be an instance of the marked GPCP. The *blocks* of the generalized instance I are the blocks of the instance (h, g) of the marked PCP defined in Chapter 4. That is, for (h, g), a block for a letter a is the minimal nontrivial solution (u, v) of the equation $h(x) = g(y)$ such that $a \leq h(x) \wedge g(y)$. We continue to denote the block for a letter a by $\beta(a) = (u, v)$. Note that also $a \leq u$ by the assumption (5.1).

In order to reduce an instance of the marked GPCP to its successor as we did in the case of the marked PCP, we need to define the reduction for the end words. For two words $s_1, s_2 \in A^*$ with $s_1 = \varepsilon$ or $s_2 = \varepsilon$, a pair (u, v) is called an *end block* (or an (s_1, s_2)-*end block*, to be precise) if $h(u)s_1 = g(v)s_2$ and for all $u_1 \leq u$ and $v_1 \leq v$, $h(u_1) \neq g(v_1)$, see Figure 5.1.

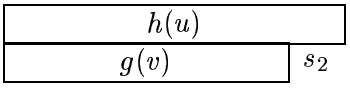

Figure 5.1: An (ε, s_2)-end block.

We proved in Lemma 4.2 that a block is unique (if it exists) for a given letter, but an end block is not necessarily unique, not even for a given letter. Let

$$E_a = \{(u, v) \mid (u, v) \text{ is an end block and } a \leq h(u) \text{ or } a \leq g(v)\}$$

be the set of all end blocks for the letter a. Note that it is possible that either u or v is ε. Recall that a subset of $\Sigma^* \times \Delta^*$ for two alphabets Σ and Δ is a *rational relation* if it is defined by a *finite transducer*, see for example Berstel [2].

Lemma 5.2. *Let $I = (\#, h, g, (s_1, s_2))$ be an instance of the marked GPCP, $s_1 = \varepsilon$ or $s_2 = \varepsilon$ and $s_1 s_2 \neq \varepsilon$, and let a be a fixed letter. The set of the end blocks E_a is a rational relation and can be effectively found. Moreover,*

(i) If a is a block letter, then E_a is finite.

(ii) If E_a is infinite, then it is a union of a finite set and finite number of sets

$$\{(xu^k, yv^k w) \mid k \geq 0\} \text{ or } \{(xu^k w, yv^k) \mid k \geq 0\}$$

according to whether $s_2 = \varepsilon$ or $s_1 = \varepsilon$, respectively, for some words u, v, x, y, w.

Proof. Without loss of generality, we may assume that $s_2 = \varepsilon$.

(i) Assume first that $a \in B$ is a block letter and that $\beta(a) = (x, y)$. Now an end block (u, v) in E_a satisfies $h(u)s_1 = g(v)$, and for all $u_1 < u$ and $v_1 < v$, $h(u_1) \neq g(v_1)$. Since the morphisms are marked and $a \leq h(u) \wedge h(x)$, clearly $u \leq x$. Since $|g(v)| > |h(u)|$, also $v = v'v''$, where $v' \leq y$ and $h(u)^{-1}g(v') \leq s_1$. There are only finitely many prefixes u of x and the words v' and v'' are unique for all prefixes u of x by the uniqueness of a block and by Lemma 4.1. This proves part (i).

(ii) Assume next that a is not a block letter. Then we have two cases:
(1) There is only a finite number of pairs (x, y) such that $h(x)$ and $g(y)$ are comparable and $a \leq h(x) \wedge g(y)$. This case goes through as the case where a is a block letter. In particular, E_a is finite.
(2) There exist infinitely many pairs (x, y) such that $h(x)$ and $g(y)$ are comparable and $a \leq h(x) \wedge g(y)$. Then the sequence (x_0, x_1) constructed by Algorithm 2 for a block is infinite, i.e., we have the case (ii) of page 36, where the same overflow is found repeatedly. This is possible only if there are words $x, y, u, v \in A^*$ and $s \in B^*$ such that $|h(u)| = |g(v)| \geq 1$ and $h(x)s = g(y)$ and $h(u)s = sg(v)$. But then after $|x| + |y|$ rounds of Algorithm 2, we have the pair $(x_0, x_1) = (xu^k u', yv^k v')$ for some k, where $u' \leq u$ and $v' \leq v$. But for $i \geq |x| + |y|$, an end block can always be written as $(xu^k u', yv^k v' w_i)$ or equivalently as $(xu'(u''u')^k, yv'(v''v')^k w_i)$ to get the desired form (here $u = u'u''$ and $v = v'v''$). Claim (ii) follows, since there are only finitely many prefixes u' and v' of u and v, and there are at most $|x| + |y|$ other potential end blocks.

The rationality of E_a follows easily from the proofs for (i) and (ii). □

We shall call $(xu^k, yv^k w)$ and $(xu^k w, yv^k)$ in Lemma 5.2 (ii) *extendible end blocks.*

Let $I = (\#, h, g, (s_1, s_2))$ be an instance of the marked GPCP. For a solution $w \in \#(A \setminus \{\#\})^*$, $h(w)s_1 = g(w)s_2$, of I,

$$w = u_1 u_2 \cdots u_{k+1} = v_1 v_2 \cdots v_{k+1}$$

is a *block decomposition* for w, if

(i) $(u_i, v_i) = \beta(a_i)$ for some letter a_i, for all $i = 1, 2, \ldots, k$ with $a_1 = \#$,

(ii) (u_{k+1}, v_{k+1}) is an (s_1, s_2)-end block.

Note that $k = 0$ is also possible in the block decomposition. In this case the first pair (u_1, v_1) is an (s_1, s_2)-end block for the letter $\#$. Therefore, we are also interested in the set $E_\#$.

Because the blocks are minimal solutions to $h(u_i) = g(v_i)$, it is easy to see that the following lemma holds.

Lemma 5.3. *Every solution $w \in \#(A \setminus \{\#\})^*$ of I has a unique block decomposition (see Figure 5.2).*

$h(w)$	$h(u_1)$	$h(u_2)$	$\cdots$	$h(u_k)$	$h(u_{k+1})s_1$
$g(w)$	$g(v_1)$	$g(v_2)$	$\cdots$	$g(v_k)$	$g(v_{k+1})$

Figure 5.2: Block decomposition of a solution w (for $s_2 = \varepsilon$)

Next we define the successors of the instances $I = (\#, h, g, (s_1, s_2))$ of the marked GPCP. Let (h', g') be the successor of the instance (h, g) of the marked PCP as defined in Chapter 4 and let (u, v) be any end block of I. Then

$$I'(u, v) = (\#, h', g', (s_1', s_2'))$$

is the *successor* of I with respect to (u, v), where (s_1', s_2') is defined as follows: if $v \preccurlyeq u$, then $s_1' = uv^{-1}$ and $s_2' = \varepsilon$, and if $u \preccurlyeq v$, then $s_1' = \varepsilon$ and $s_2' = vu^{-1}$. If u and v are not suffix comparable, i.e., $u \not\preccurlyeq v$ or $v \not\preccurlyeq u$, then $I'(u, v)$ is not defined.

Note that for $(u, v) \in E_\#$, the successor $I'(u, v)$ is defined if and only if $u = v$, since $\#$ is a special symbol occurring in $h(A)$ and $g(A)$ only as the first letter of $h(\#)$ and $g(\#)$. Moreover, in this case $u = v$ is a solution of I.

Lemma 5.4. *An instance $I = (\#, h, g, (s_1, s_2))$ has a solution if and only if the successor $I'(u, v) = (\#, h', g', (s_1', s_2'))$ has a solution for some end block (u, v). Moreover, each solution w to I can be written as $w = h'(w')u = g'(w')v$, where w' is a solution of $I'(u, v)$.*

Proof. Assume first that I has a solution w with the block decomposition

$$w = u_1 u_2 \cdots u_{k+1} = v_1 v_2 \cdots v_{k+1},$$

where $(u_i, v_i) = \beta(a_i)$ for $1 \leq i \leq k$, and (u_{k+1}, v_{k+1}) is an end block. Clearly $u_{k+1} \preccurlyeq v_{k+1}$ or $v_{k+1} \preccurlyeq u_{k+1}$. If the first case holds, then $s_1' = \varepsilon$, $s_2' = v_{k+1}u_{k+1}^{-1}$ and $I'(u, v) = (\#, h', g', (s_1', s_2'))$. Now

$$h'(a_1 \cdots a_k) = wu_{k+1}^{-1} = g'(a_1 \cdots a_k)s_2',$$

i.e., the successor $I'(u_{k+1}, v_{k+1})$ has a solution $w' = a_1 \cdots a_k$ and, moreover, $w = h'(w')u_{k+1} = g'(w')v_{k+1}$. The case $v_{k+1} \preccurlyeq u_{k+1}$ is symmetric to the above.

Assume then that

$$I'(u, v) = (\#, h', g', (s'_1, s'_2))$$

has a solution w', i.e., $h'(w')s'_1 = g'(w')s'_2$. Then also $h'(w')u = g'(w')v$ and by the definition of the end blocks and Lemma 4.4(i),

$$h(h'(w')u)s_1 = h(h'(w'))h(u)s_1 = g(g'(w'))g(v)s_2 = g(g'(w')v)s_2,$$

therefore $h'(w')u = g'(w')v$ is a solution of I. □

For $(u, v) \in E_\#$, $w' = \varepsilon$ in the previous theorem. Clearly it is of no use to construct the successors of I for the special case $(u, u) \in E_\#$, since in these cases we have already found a solution of I. We shall return to this in Lemma 5.10.

5.2 Decidability of the marked GPCP

In this section we prove that the marked GPCP is decidable. There are two important steps in our proof. First, we prove that the number of different successors is finite and, second, that if there is a solution, then the length of the new end words (s'_1, s'_2) is bounded by a constant.

5.2.1 Extendible end blocks

By Lemma 5.4, we can reduce an instance I to its successors for all end blocks. The problem in this approach is that by Lemma 5.2, I potentially has infinitely many successors. We shall next show that the extendible end blocks also reduce to a finite number of successors.

Let $I = (\#, h, g, (s_1, s_2))$ and assume that there exist the successors

$$I'(xu^k w, yv^k) = I'_k \text{ for } k \geq 0 \tag{5.5}$$

with the morphisms $h', g' \colon (A')^* \to A^*$ as defined for the successor of (h, g). The successors $I'(xu^k, yv^k w)$ are treated analogously.

Lemma 5.5. *Let I'_k, for $k \geq 0$, be as in the above and assume that $|u| = |v|$. Then there are only finitely many distinct successors I'_k and they can be effectively found.*

Proof. The successor I'_k with respect to $(xu^k w, yv^k)$ is defined only if either $yv^k \preccurlyeq xu^k w$ or $xu^k w \preccurlyeq yv^k$. In the first case, $s'_1 = xu^k w(yv^k)^{-1}$ and $s'_2 = \varepsilon$. It is straightforward to see that $|xu^k w(yv^k)^{-1}| = |xw| - |y|$ for all $k \geq 0$, if $xu^k w(yv^k)^{-1}$ is defined.

Let ℓ be the least number such that $|yv^\ell| > |w|$ and $yv^\ell \preccurlyeq xu^\ell w$. Then $xu^k w(yv^k)^{-1} = xu^\ell w(yv^\ell)^{-1}$ for all $k \geq \ell$, since these have the same length and they are both prefixes of xu^k. This implies that $I'_k = I'_\ell$ for all $k \geq \ell$. Therefore, there exist at most ℓ different successors I'_k, and this ℓ can be effectively found.

The second case, $xu^k w \preccurlyeq yv^k$, is similar. □

Lemma 5.6. *Let I'_k, $k \geq 0$, be as in* (5.5). *For all letters $a \in A'$, $h'(a) \wedge xu = \varepsilon$ and $g'(a) \wedge yv = \varepsilon$.*

Proof. Notice first that if $b \leq xu$ and $c \leq yv$ for some $b, c \in A$, then $b \leq g(c)$, since $h(xu)$ and $g(yv)$ are comparable (and $b \leq h(b) \leq h(xu)$). It follows that no $h'(w_1)$ begins with the letter b and no $g'(w_2)$ begins with the letter c. Namely, if $h'(b) = bz_1$ and $g'(d) = cz_2$ for some words z_1 and z_2 for $d \in A$, then by the definition of h' and g', $\beta(b) = (bz_1, cz_2)$ for I, since $b \leq h(bz_1) \wedge g(cz_2)$, but this contradicts Lemma 5.2. □

Lemma 5.7. *Assume that the successors I'_k, $k \geq 0$, are as in* (5.5), *and that $|u| \neq |v|$. Then there are only finitely many k's such that I'_k is nontrivial, and these instances can be effectively found.*

Proof. We can assume that $k \geq 1$. To prove the claim it suffices to show that for all $f \in A'$, there are only finitely many k's such that I'_k has an end block for f such that the words of the end block are suffix comparable. For this purpose, assume that I'_k has an end block (u_1, v_1) for $f \in A'$. Recall that I'_k is defined only if $yv^k \preccurlyeq xu^k w$ or $xu^k w \preccurlyeq yv^k$. In the first case, $s'_1 = xu^k w(yv^k)^{-1}$ and $s'_2 = \varepsilon$. In the second case $s'_1 = \varepsilon$ and $s'_2 = yv^k(xu^k w)^{-1}$. Consider now the second case, the first one is analogous. Note that s'_2 depends on k. If $|u| > |v|$, then $xu^k w \preccurlyeq yv^k$ can be true only for finitely many k, and so we can assume $|u| < |v|$ as well.

Now $h'(u_1) = g'(v_1)s'_2$. Let $u_0 \leq u_1$ be such that $h'(u_0)$ covers $g'(v_1)$, that is, u_0 is the minimum word such that $h'(u_0) = g'(v_1)z$ for some z. In particular, $z \leq s'_2 \leq yv^k$, and hence, by Lemma 5.6, $z \wedge g'(d) = \varepsilon$ for all $d \in A'$. Note that $z \neq \varepsilon$, since otherwise $\beta(f) = (u_0, v_1)$, which is not allowed by the definition of the end block. It follows that v_1 is not a prefix in any other end block, since by Lemma 4.1, $h(u_0)$ is the unique cover of $g'(v_1)$. Note that it now follows that the word v_1 can be effectively found by searching the possible (end) block for f, and once the overflow of g' is a prefix of yv^k, the word v_1 has been found.

We will finally show that the word $u' = u_0^{-1}u_1$ (or actually u_1) and the integer k are unique if they exist. The word u' is constructed letter by letter defining $u'_0 = \varepsilon$ and $u'_{j+1} = u'_j a$, where a is the letter for which $h'(u_0)h'(u'_j a) \leq g'(v_1)y^k v$ for all large enough k. Because h' is marked, such a letter a is always unique, if it exists.

Assume that b is the first letter of xu^kw. Now xu^kw is a suffix of yv^k and, by Lemma 5.6, no $h'(d)$ begins with b. Therefore, if u' exists, it is unique, since $(h'(u_0)h'(u'))^{-1}g'(v_1)y^kv = bz'$ for some word z'. If u' exists, then k is unique and easily determined, since necessarily

$$|h'(u_0)h'(u')xu^kw| = |g'(v_1)y^kv|$$

and $|u| \neq |v|$. □

Note that if there exists a k in Lemma 5.7 such that I'_k is nontrivial, then the size of the alphabet decreases by Lemma 5.6 while moving from I'_k to its successor. Indeed, if there is an end block for a letter f in I'_k, then there cannot exist a block for f.

From Lemmata 5.5 and 5.7 we obtain

Corollary 5.1. *For an extendible end block, the instances as in* (5.5) *consist of only finitely many different successors.*

5.2.2 Simple cases

Let

$$F' = \{(s'_1, s'_2) \mid (\#, h', g', (s'_1, s'_2)) \text{ is a successor of } I\}$$

be the set of the end pairs of the successors of an instance I. We introduce a collective notation

$$\mathcal{I}' = (\#, h', g', F') \tag{5.6}$$

to stand for all the successors of I. By Lemma 5.4, I has a solution if and only if one of the successors (5.6) has. By Corollary 5.1, we can always assume that F' is a finite set. Thus, we obtain a chain of sets of successors reducing the original instance $I_0 = (\#, h_0, g_0, (s_1, s_2))$ to its successors $\mathcal{I}_1 = (\#, h_1, g_1, F_1)$, then all these reduce to $\mathcal{I}_2 = (\#, h_2, g_2, F_2)$, etc. We shall next prove that if eventually some successors $\mathcal{I}_i = (\#, h_i, g_i, F_i)$ have very simple instances (h_i, g_i) (i.e., $|A_i \setminus \{\#\}| = 1$ or the suffix complexity is zero), then we can decide for each instance in $(\#, h_i, g_i, F_i)$ whether it has a solution or not.

We consider first the unary case. Note that the next lemma holds for all unary instances of the GPCP, not only for the marked instances.

Lemma 5.8. *The GPCP is decidable in the unary case.*

Proof. Let $I = ((p_1, p_2), h, g, (s_1, s_2))$ be an instance of the GPCP, where $h, g\colon \{a\}^* \to B^*$. We may clearly assume that $h(a) \neq \varepsilon$. Then

$$p_1h(a^k)s_1 = p_2g(a^k)s_2 \tag{5.7}$$

for some k only if

$$k(|h(a)| - |g(a)|) = |p_2| + |s_2| - |p_1| - |s_1|.$$

First, if $|h(a)| \neq |g(a)|$, then k is unique and it can be effectively found by the previous equality. Assume $|h(a)| = |g(a)|$ and $p_1 = \varepsilon$. Let t be the least integer such that $|p_1 h(a)^t| > |p_2|$. Now $(p_1 h(a)^k)^{-1}(p_2 g(a)^k) = (p_1 h(a)^t)^{-1}(p_2 g(a)^t$ for all $k \geq t$, and $p_1 h(a)^k < p_2 g(a)^k$ if and only if $p_1 h(a)^t < p_2 g(a)^t$. Therefore, we have only t different cases to check for a solution. □

Since in our instances $(\#, h, g, (s_1, s_2))$ we always assume that the special letter $\#$ is included in domain alphabet, we obtain the following corollary.

Corollary 5.2. *The marked GPCP is decidable for any instance* $(\#, h, g, (s_1, s_2))$, *where* $h, g\colon A^* \to B^*$ *and* $|A \setminus \{\#\}| = 1$.

For the case where the suffix complexity is zero, we obtain

Lemma 5.9. *It is decidable whether an instance of the marked GPCP, where the morphisms have suffix complexity zero, has a solution.*

Proof. Let $I = (\#, h, g, (s_1, s_2))$, where $h, g\colon A^* \to B^*$, and $\sigma(I) = 0$. Therefore, for all $a \in A$, $|h(a)| = 1$. Thus, there is a solution w to I only if $|s_1| - |s_2| = 0$, and hence there is a solution if and only if $h(\#) = \# = g(\#)$ and $s_1 = s_2$. □

In our solution we do not construct the successors for the end blocks in $E_\#$, but check at each step whether there is $(u, v) \in E_\#$ such that $u = v$. Note that the end block $(\#, \#)$ is also possible in $E_\#$, since we also allow empty solutions for the GPCP.

Lemma 5.10. *For each instance of the marked GPCP, we can effectively check whether there is an end block* $(u, v) \in E_\#$ *such that* $u = v$.

Proof. If $E_\#$ is finite, then the claim is obvious, see Lemma 5.2. Therefore we assume that $E_\#$ is infinite, i.e., there is an extendible end block in $E_\#$. Assume furthermore that there is an extendible end block of the form $(xu^k w, yv^k)$, the case $(xu^k, yv^k w)$ is analogous. Now there is a solution of the instance among the pairs $(xu^k w, yv^k)$, for all $k \geq 0$, if and only if $xu^k w = yv^k$ for some k. We have again two cases to consider depending on whether $|u| = |v|$ or not.

Assume first that $|u| \neq |v|$. Then $|xu^k w| = |yv^k|$ for a unique k and this k can be effectively determined. Therefore, we have to check only whether $xu^k w = yv^k$ for this unique k.

Assume next that $|u| = |v|$. If there is a solution, then $|x| \leq |y|$, since $|xw| = |y|$. Let t be the least integer such that $|xu^t| > |y|$. Now for all $k \geq t$, $(xu^k)^{-1}(yv^k) = (xu^t)^{-1}(yv^t)$ and $(xu^k w, yv^k)$ is a solution if and only if $(xu^t w, yv^t)$ is a solution. Therefore, we have only t different pairs to check for a solution. □

5.2.3 Cycling instances

We know now that in the above simple cases we can solve the original problem. Otherwise, by Lemma 4.6, there are integers d and n_0 such that $(h_{i+d}, g_{i+d}) = (h_i, g_i)$ for each $i \geq n_0$, i.e., the instances (of the marked PCP) form an ultimately periodic sequence. Clearly, to show that the marked GPCP is decidable, it suffices to show how to solve these *cycling instances*.

By a *successor sequence* we mean a sequence

$$(\#, h_0, g_0, (s_1^{(0)}, s_2^{(0)})), \ldots, (\#, h_i, g_i, (s_1^{(i)}, s_2^{(i)})), \ldots \tag{5.8}$$

of instances of the marked GPCP such that each $(\#, h_{i+1}, g_{i+1}, (s_1^{(i+1)}, s_2^{(i+1)}))$ is a successor of $(\#, h_i, g_i, (s_1^{(i)}, s_2^{(i)}))$ and the instance $(\#, h, g, (s_1, s_2)) = (\#, h_0, g_0, (s_1^{(0)}, s_2^{(0)}))$ is the original instance.

We now show how to treat the cycling instances, i.e., such that for all successor sequences as in (5.8), there exists a d, where $(h_i, g_i) = (h_{i+d}, g_{i+d})$ for all $i \geq 0$. The first instance in (5.8) is denoted by I_0. We can assume that the cycle begins at $i = 0$, that is, $(h_0, g_0) = (h_d, g_d)$, since if this case is decidable, then the cycling case is also decidable in general.

Such an instance I_0 is called *cycling instance*, and d is the *length of the cycle*. Note that the instances in a cycle can all be different, since the end pairs $(s_1^{(i)}, s_2^{(i)})$ can be different. We assume only that the pairs of morphisms are the same modulo d.

Assume that I_0 is a cycling instance. Notice that if $\mathcal{I}_i = (\#, h_i, g_i, F_i)$ is the set of the ith members in (all of) the successor sequences, we can always assume that

(A) the pair (h_i, g_i) has a block for the letter $\#$, and
(B) $s_1 s_2 \neq \varepsilon$ for each $(s_1, s_2) \in F_i$.

The condition (A) necessarily holds, since I_0 is a cycling instance and the domain alphabet does not decrease anymore. If (B) is not satisfied by an instance, then that instance reduces to an instance of the marked PCP where the solution has to belong to $\#(A \setminus \{\#\})^*$, which is decidable by Theorem 4.3.

Notice that the size of the alphabet does not decrease in a cycle, and therefore there is a block for each letter in the domain alphabet. In particular, we state

Lemma 5.11. *Let I_0 be a cycling instance as above. Then there cannot be extendible end blocks in any successor sequence of I_0.*

Proof. If there is an extendible end block, then the alphabet size decreases, a contradiction, since I_0 is cycling and the morphisms are the same modulo d. □

Lemma 5.12. *Assume that I_0 is a cycling instance as in* (5.8) *and that a solution to I_0 exists. Then we have two cases:*

(i) If $h_0(\#) = \# = g_0(\#)$, then the minimal solution of I_0 is $\#w$, where the initial letter a of w satisfies $h_0(a) \neq g_0(a)$. Hence, also $h_i(a) \neq g_i(a)$ for all $i \geq 0$.

(ii) If $h_0(\#) \neq g_0(\#)$, then the minimal solution beginning with # does not have a solution of the PCP as a prefix, i.e., if $\#w$ is the minimal solution of some instance $(\#, h_i, g_i, (s_1^{(i)}, s_2^{(i)}))$ in (5.8)*, then for all $w' < w$, $h_i(\#w') \neq g_i(\#w')$.*

Proof. These claims follow by the facts that the pairs of morphisms are in the cycle and that the minimal solutions of the instance (h_0, g_0) of the marked PCP are of length one. □

Hereafter we will assume that $h_0(\#) \neq g_0(\#)$, since by case (i) of Lemma 5.12, the case where $h_0(\#) = \# = g_0(\#)$ is equivalent to deciding the instances $(\#, h_0, g_0, (s_1^{(0)}, s_2^{(0)}))$, where $h_0(\#) = \#h_0(a)$ and $g_0(\#) = \#g_0(a)$ for each a such that $h_0(a) \neq g_0(a)$.

In order to design a decision method for the marked GPCP, we need to find the possible solutions in the cycling instances. For this, we would like to have an upper bound for the lengths of the new end blocks in the cycle (5.8). We demonstrate that there exists a constant L such that, *if a solution exists*, then a *minimal solution* is found in some sequence (5.8) where the end words are shorter than L. Moreover, this constant can be effectively found, and hence the decidability of the marked GPCP will follow.

In what follows, we assume that $I = (\#, h, g, (s_1, s_2))$ has a minimal solution beginning with # and I does not have a solution of the PCP as a prefix. It is clear that a minimal solution is unique, since the morphisms are marked. Consequently, I has a unique end block (u, v) in the block decomposition of the minimal solution. It follows that there exists a unique successor sequence $I_0, I_1, \ldots$ of instances such that

$$I_{i+1} = I_i'(u_i, v_i), \tag{5.9}$$

where (u_i, v_i) is the end block of the minimal solution (beginning with #) of I_i. This successor sequence is called *the branch of the minimal solutions*. Note that we cannot determine beforehand the end block of the minimal solution, but the desired upper bound will be obtained.

Let the sequence $I_i = (\#, h_i, g_i, (s_1^{(i)}, s_2^{(i)}))$ be the branch of the minimal solutions and w_i be the minimal solution of I_i. Recall that we permanently assume that $s_1^{(i)} s_2^{(i)} \neq \varepsilon$ and also that $g_0(\#) \neq h_0(\#)$, which implies that $g_i(\#) \neq h_i(\#)$ for each i.

Lemma 5.13. *Let w_i be the minimal solution of I_i beginning with # and let $(h_i, g_i) = (h_{i+d}, g_{i+d})$ for each i. Then $w_{i+d} \leq w_i$ and $|w_{i+d}| \leq |w_i| - d$ for each i.*

Proof. The instances

$$I_i = (\#, h_i, g_i, (s_1^{(i)}, s_2^{(i)})) \text{ and } I_{i+d} = (\#, h_i, g_i, (s_1^{(i+d)}, s_2^{(i+d)}))$$

share the marked morphisms, and therefore $w_i \leq w_{i+d}$ or $w_{i+d} \leq w_i$, since the minimal solution cannot have a solution of the PCP as a prefix (recall that $h_i(\#) \neq g_i(\#)$).

If w is a minimal solution of some instance I_i in the branch of minimal solutions, then by Lemma 5.4 there is a solution w' to the successor I_{i+1} of I_i such that $w = h_{i+1}(w')u = g_{i+1}(w')v$, where (u, v) is an end block of I_i. Notice that w and w' begin with the same letter. Since $s_1^{(i)} s_2^{(i)} \neq \varepsilon$, then also $uv \neq \varepsilon$ and consequently $|w| > |w'|$, because the morphisms are nonerasing.

Hence, in the sequence I_i, $|w_{i+1}| + 1 \leq |w_i|$. Inductively, $|w_{i+t}| + t \leq |w_i|$ for all t.

Since $h_i(\#) \neq g_i(\#)$, i.e., there are no solutions for the instance $(h_i, g_i)(= (h_{i+d}, g_{i+d}))$ of the PCP beginning with #, and w_i and w_{i+d} begin with same letter, necessarily $w_{i+d} \leq w_i$. This proves the claim. □

As a byproduct we obtain

Lemma 5.14. *If an instance of the marked GPCP occurs twice in a successor sequence, it has no solutions.*

Proof. By the proof of the previous lemma, the length of the minimal solution decreases strictly when a cycle is traversed once. □

An end block (u, v) of an instance $I = (\#, h, g, (s, \varepsilon))$ satisfies the equation

$$h(u)s = g(v).$$

If (u, v) is an end block in the block decomposition of a solution, then necessarily $u = s'v$ or $v = s'u$ for some word s', and I', which is a successor of I, has the end pairs (s', ε) or (ε, s'), respectively.

Lemma 5.15. *Let $I_i = (\#, h_i, g_i, (s_1^{(i)}, s_2^{(i)}))$ be the branch of the minimal solutions of a cycling instance with a cycle of length d. Let also w_i be the minimal solution of I_i. Then $h_i(w_{i+d})s_1^{(i+d)} = g_i(w_{i+d})s_2^{(i+d)}$ is a prefix of both $h_i(w_i)$ and $g_i(w_i)$.*

Proof. It suffices to consider $i = 0$; the proof is the same for all other values. Recall that $s_1^{(t)} s_2^{(t)} \neq \varepsilon$ for each t. By Lemma 5.13, $w_d \leq w_0$. Therefore $h_0(w_d) \leq h_0(w_0)$ and $g_0(w_d) \leq g_0(w_0)$.

We shall next prove that $|h_0(w_d)s_1^{(d)}| \leq |h_0(w_0)|$. By Lemma 5.4,

$$w_0 = h_1(h_2(\cdots h_{d-1}(h_d(w_d)u_{d-1})\cdots u_2)u_1,$$

where (u_m, v_m) is the end block of the minimal solution of the instance I_m, for $1 \leq m \leq d-1$. Therefore, since $h_0 = h_d$ and the morphisms h_i are nonerasing,

$$|h_0(w_0)| \geq |h_0(w_d)| + \sum_{i=1}^{d-1} |u_i| \geq |h_0(w_d)| + |s_1^{(d)}|,$$

where the last inequality follows from $|s_1^{(d)}| \leq |u_{d-1}|$. Therefore $|h_0(w_d)s_1^{(d)}| \leq |h_0(w_0)|$, and similarly we can prove that $|g_0(w_d)s_2^{(d)}| \leq |g_0(w_0)|$.

Without loss of generality, we can assume that $s_2^{(d)} = \varepsilon$. Then, using Lemma 5.13, we obtain

$$h_0(w_d)s_1^{(d)} = g_0(w_d)s_2^{(d)} = g_0(w_d) \leq g_0(w_0).$$

Since $h_0(w_0)$ and $g_0(w_0)$ are comparable and $|h_0(w_d)s_1^{(d)}| \leq |h_0(w_0)|$, necessarily $h_0(w_d)s_1^{(d)} \leq h_0(w_0)$, which proves the claim.

□

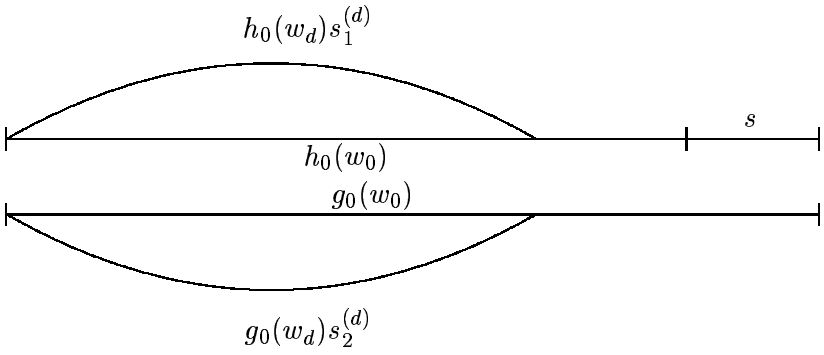

Figure 5.3: Prefix property of Lemma 5.15

The previous lemma is used in the proofs of our last two lemmata, which give an upper bound for the size of the end blocks in the branch of the minimal solutions.

We defined in Section 4.1.1 the notation of a g-cover for a marked morphism g. We called a word u a g-cover of w if $w \leq g(u)$ and $g(u') < w$ for all $u' < u$. Note that the g-cover of w is unique, since g is marked.

Let (h, g) be an instance of the marked PCP. For an occurrence of a word u in $g(w)$, its *g-block cover* is a word $z = g(v_1)g(v_2)\cdots g(v_k)$ such that

(i) $v_1v_2\cdots v_k$ is a factor of w,

(ii) u is a factor of z,

(iii) the occurrence of u is not within the factors $g(v_2)\cdots g(v_k)$ or $g(v_1)\cdots g(v_{k-1})$ of $g(w)$,

(iv) for each i the word v_i is a block word, i.e., $(u_i, v_i) = \beta(a_i)$ for some letter a_i and word u_i.

Note that the g-block cover is defined for occurrences of words in $g(w)$. Therefore, if there are two occurrences of u in $g(w)$, then they may have different g-block covers. On the other hand, a g-block cover for an occurrence of a factor u in $g(w)$ is unique. Hence we can define the integer k (from (i)) to be the *g-cover length* of the occurrence of u (in $g(w)$).

The h-block cover is defined analogously.

In what follows, we shall concentrate on the block cover and cover lengths of the end words s_1 and s_2 as they occur as factors in $h(w)$ and $g(w)$.

Lemma 5.16. *Let $I_i = (\#, h_i, g_i, (s_1^{(i)}, s_2^{(i)}))$ be the branch of the minimal solutions of a cycling instance having cycle of length d, and w_i be the minimal solution of I_i. Then the h_i- and g_i-block covers of $s_1^{(i)}$ and $s_2^{(i)}$ exist in $g_i(w_{i-d})$ and $h_i(w_{i-d})$, for all $i \geq d$.*

Proof. By Lemma 5.15, for $i \geq d$, the words $s_1^{(i)}$ and $s_2^{(i)}$ are factors in $h_i(w_{i-d})$ and $g_i(w_{i-d})$, respectively, since $h_i(w_i)s_1^{(i)}$ and $g_i(w_i)s_2^{(i)}$ are prefixes of $h_i(w_{i-d})$ and $g_i(w_{i-d})$. From these observations, the claim follows. □

Note that in the next lemma the occurrences of $s_1^{(i)}$ and $s_2^{(i)}$ in $h_i(w_{i-d})$ and $g_i(w_{i-d})$ considered are exactly the suffixes of $h_i(w_i)s_1^{(i)}$ and $g_i(w_i)s_2^{(i)}$, where, by Lemma 5.15, these words are prefixes of $h_i(w_{i-d})$ and $g_i(w_{i-d})$. Note also that $s_1^{(i)}$ and $s_2^{(i)}$ are factors of both $h_i(w_{i-d})$ and $g_i(w_{i-d})$.

Lemma 5.17. *Let $I_i = (\#, h_i, g_i, (s_1^{(i)}, s_2^{(i)}))$ be the branch of the minimal solutions of a cycling instance having cycle of length d, and w_i be the minimal solution of I_i. For all $i \geq d$,*

(i) If $s_1^{(i)} \neq \varepsilon$, then the h_{i+1}-cover lengths of $s_1^{(i+1)}$ and $s_2^{(i+1)}$ (in $h_{i+1}(w_{i+1-d})$) are at most the g_i-cover length of $s_1^{(i)}$ (in $g_i(w_{i-d})$).

(ii) If $s_2^{(i)} \neq \varepsilon$, then the g_{i+1}-cover lengths of $s_1^{(i+1)}$ and $s_2^{(i+1)}$ (in $g_{i+1}(w_{i+1-d})$) are at most the h_i-cover length of $s_2^{(i)}$ (in $h_i(w_{i-d})$).

Proof. By Lemma 5.16, the block covers exist. We will prove only case (i), the other one being similar. To simplify the notations, we denote $I = I_i = (\#, h, g, (s, \varepsilon))$ and $I' = I_{i+1}$. Now either $I' = (\#, h', g', (s', \varepsilon))$ or $I' = (\#, h', g', (\varepsilon, s'))$. We have to show that in both cases the h'-cover length of s' is at most the g-cover length of s.

Assume that (u, v) is the end block of the minimal solution w of I. Let a be the first letter of u. Since $h(u)s = g(v)$ and $u \preccurlyeq v$ or $v \preccurlyeq u$, the above two cases can be divided as follows:
(1) Assume that $u = s'v$, which is the case $I' = (\#, h', g', (s', \varepsilon))$. Then $|s'| \leq |u|$. Because there is a block for the letter a in the instance I, necessarily $s' \leq h'(a)$, i.e., the h'-cover length of s' is 1 (see Figure 5.4).

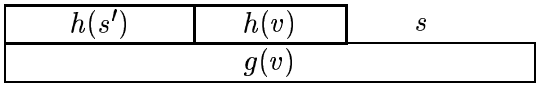

Figure 5.4: Picture of case (1)

(2) Assume then that $v = s'u$, and so $I' = (\#, h', g', (\varepsilon, s'))$. We observe first that the g-cover length of the word $g(s')$ is at most that of the word s. This is clear, because $g(s')$ shares with s every one of its block factors $g(v_i)$ (including the first one, since as in case (1), $h(u)$ is covered by a single g-block longer than $h(u)$ (see Figure 5.5 for an illustration)).

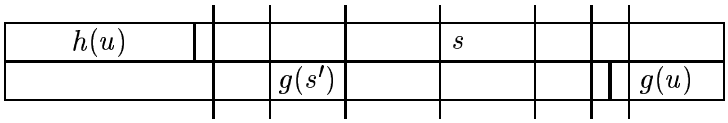

Figure 5.5: Block covering of $h(u)s = g(s')g(u)$. The vertical lines illustrate the block covering.

We show then that the h'-block cover of s' is not longer than the g-block cover of $g(s')$, from which the claim follows. Let w' be the minimum solution of I'. Then the word $w = h'(w') = g'(w')s'$ satisfies

$$h(w)g(s') = hh'(w')g(s') = gg'(w')g(s') = g(w),$$

and consequently w is a prefix of the minimal solution of I. To show that the h'-block cover of s' is not longer than the g-block cover of $g(s')$, it is sufficient to show that the block borderlines in $h'(w') = g'(w')s'$ that cut s' can be mapped injectively to the block borderlines in $h(w)g(s') = g(w)$ that cut $g(s')$ (see Figure 5.6). Let $y' \leq w'$ be a word that determines a block borderline in $h'(w') = g(w')s'$ that cuts s'. That is, $h'(y') = g'(z')$ for some word z' and $g'(w') < h'(y')$. Then $z' = w'x'$ for some x' that satisfies

$g'(x') \leq s'$. Now the word $y = g'(z')$ is a prefix of w, since

$$g(y) = g(g'(z')) = g(g'(w')g'(x')) \leq g(g'(w')s') = g(w)$$

and g is marked. But y also determines a block borderline in the word $g(w) = h(w)g(s')$, since $g(y) = g(g'(z')) = h(h'(z'))$. This borderline cuts $g(s')$, because

$$\begin{aligned} g(y) &= g(g'(z')) = h(h'(z')) \\ &= h(h'(w'))h(h'(x')) = h(w)h(h'(x')), \end{aligned}$$

and hence $h(w) \leq g(y)$. Notice finally that the word y determines z' uniquely, since g is injective and z' determines y' by $g'(z') = h'(y')$. (Recall that h is injective.)

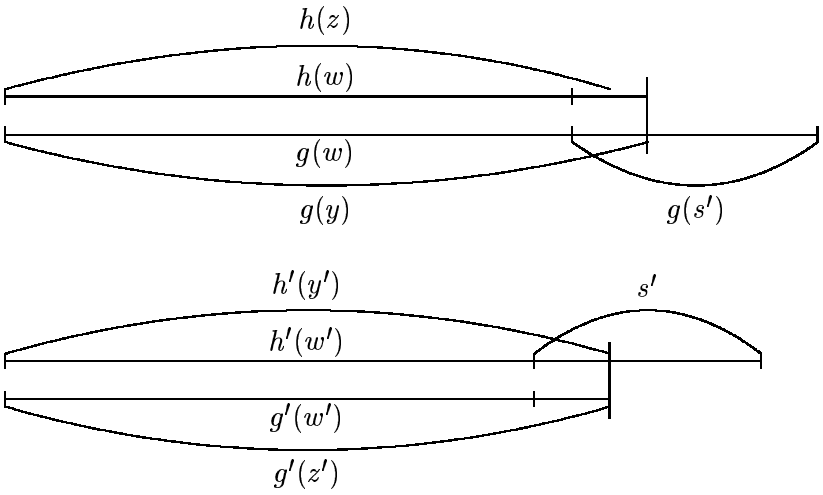

Figure 5.6: Relation between the block borderlines of $g(s')$ and s'

□

The previous lemma gives us a tool for recognizing instances which *are not* in the branch of the minimal solutions. Let I_0 be a cycling instance with cycle of length d and consider *all* the instances $\mathcal{I}_d$ found by the first d reductions. If I_0 has a solution then there is a unique $I \in \mathcal{I}_d$ in the branch of the minimal solutions.

Let M be the maximal g- or h-cover length of all the end words s_1 and s_2 in $\mathcal{I}_d$. Maximal here means that we use the upper bound found by checking all possible g- and h-cover lengths for the end words. Note that we are forced to use this upper bound, since we do not know beforehand the minimal solutions w_i, not even if they exist. It now follows, by Lemma 5.17,

that in the branch of the minimal solutions the g_i- or h_i-cover length is always less than or equal to M.

For the sequence of a cycling instance, the suffix complexity is the constant $\sigma(I_0)$, and since the blocks of an instance I_i are the images of the successor I_{i+1}, the block length can never be more than $\sigma(I_0) + 1$. By the previous lemma, we have

Corollary 5.3. *Let $I_0, \ldots, I_i, \ldots$ be the branch of the minimal solutions of a cycling instance with a cycle of length d. For each $i \geq d$, the end words of I_i are not longer than $M(\sigma(I) + 1)$.*

Now we are ready to prove Theorem 5.1.

Proof of Theorem 5.1. It remains to be shown how to solve the marked GPCP for the cycling instances I_0. A cycling instance has a block for each letter. In particular, there are no extendible end blocks and only finitely many successors. The successor relation naturally defines a tree $\mathcal{T}$ having I_0 as the root, all the successors of I_0 as the vertices and the pairs (I, I') as the edges.

The decision procedure is based on constructing $\mathcal{T}$ partially by first inserting the vertices having depth (the distance from the root) at most d and then computing the number M, the maximal cover length of the end words of instances at depth d. For all vertices we check whether there is an end block $(u, v) \in E_{\#}$ such that $u = v$ as in Lemma 5.10, and for each vertex $I = (\#, h, g, (s_1, s_2))$ that has $s_1 s_2 = \varepsilon$, we check if it has a solution or not, by Theorem 4.3. If some such vertex I has no solution, then I and all the successors of I are removed. On the other hand, if some such I has a solution, then, by Lemma 5.4, I_0 also has a solution and the procedure stops.

For the vertices having depth greater than d, the (partial) construction of $\mathcal{T}$ is more specific: only the successors $I = (\#, h, g, (s_1, s_2))$ that satisfy $|s_1 s_2| \leq M(\sigma(I_0) + 1)$ are inserted. By Corollary 5.3, the branch of the minimal solutions is included in the partial construction.

By Corollary 5.3, there are only finitely many instances to be inserted, so each path (successor sequence) in the partially constructed $\mathcal{T}$ will eventually contain an instance twice. Thus I_0 has no solution by Lemma 5.14, unless one of the finitely many vertices I_i has a solution $(u, u) \in E_{\#}$ for some u. □

5.3 The algorithm

We have proved in Theorem 5.1 that the generalized Post Correspondence Problem is decidable for marked morphisms. Below we summarize the decision method in our algorithm. We do not present the procedure in details, but we state only the most significant steps.

The decision procedure achieved reduces an instance of the GPCP to finitely many simpler equivalent instances. By continuing this reduction to

each reduced instance we create a successor tree where the decision is made in each path separately according to the following six rules:

(i) If we reach an end block (u, u), then solve the instance of the marked PCP (Theorem 4.3).

(ii) If we obtain an end block $(\#u, \#u) \in E_{\#}$, then there is a solution for I (Theorem 5.10).

(iii) If we obtain an instance where the morphisms are unary or the suffix complexity is zero, then the existence of a solution can be decided by Corollary 5.2 and Lemma 5.9, respectively.

(iv) If we reach an instance which already occurred in the path, then instances in this path do not have a solution (Lemma 5.14).

(v) If the cover lengths of the end words grow over the bound $M(\sigma(I)+1)$, then the instance is not in the branch of minimal solutions, and we can omit the rest of the path (Corollary 5.3).

(vi) If there are no end blocks, then there is no solution.

Note that the complexity of our algorithm is enormous. Actually, it is at least double exponential, since the decision procedure for the marked PCP is exponential and there may be exponentially many instances in the tree of successors in the marked GPCP.

5.4 The binary PCP and GPCP

Our proof for the decidability of the marked GPCP, Theorem 5.1, yields a new proof for the decidability of the binary PCP and the binary GPCP. Here we shall describe the reductions from the binary cases to the binary marked GPCP. Note that the proof of the marked binary GPCP would be somewhat easier if we concentrate only on the binary case (see Halava, Harju and Hirvensalo [18]).

Let (h, g), where $h, g\colon A^* \to B^*$, be an instance of the binary PCP, i.e., we assume that $|A| = 2$. As mentioned, this problem was proved to be decidable by Ehrenfeucht, Karhumäki and Rozenberg [5]. The basic idea in [5] is that each instance (h, g) of the binary PCP is either

(i) periodic, i.e., $h(A^*) \subseteq u^*$, where $u \in B^*$, see Section 3.7, or

(ii) it can be reduced to an equivalent instance of the binary GPCP with marked morphisms.

Then it is proved in [5] that both of these two cases are decidable.

Here he shall first present the reduction from the binary PCP to the marked GPCP. The decidability of the case (i) was proved in Theorem 3.7. In [5] the proof of case (ii) is by case analysis and it is rather long. A somewhat shorter proof is presented in [18], where the proof of the marked GPCP of this chapter is considered in the binary case.

Note that in the PCP and GPCP we may always assume that the image alphabet B is binary, since any B can be injectively encoded to $\{0,1\}^*$. For example, if $B = \{b_1, b_2, \ldots, b_m\}$, then $\varphi\colon B \to \{0,1\}^*$, where

$$\varphi(b_i) = 01^i, \quad \text{for all } 1 \leq i \leq m,$$

is such an encoding. Therefore in the binary case we shall assume that $A = B = \{0,1\}$.

5.4.1 From binary PCP to marked GPCP

Let $h\colon \{0,1\}^* \to \{0,1\}^*$ be a morphism that is not periodic. In particular, h is nonerasing. Define the mapping $h^{(1)}$ by

$$h^{(1)}(x) = \operatorname{pref}_1(h(x))^{-1}\big(h(x)\operatorname{pref}_1(h(x))\big) \qquad \text{for } x = 0, 1.$$

In other words, the images of $h^{(1)}$ are the cyclic shifts of the images of h. Now define recursively $h^{(i+1)} = (h^{(i)})^{(1)}$. Clearly,

$$h^{(i)}(x) = \operatorname{pref}_j(h(x))^{-1}\big(h(x)\operatorname{pref}_j(h(x))\big),$$

where $j \leq |h(x)|$ and $j \equiv i \pmod{|h(x)|}$.

For any two words $u, v \in A^*$, it is well known that $uv = vu$ if and only if u and v are powers of a common word. It follows from this that the maximum common prefix of $h(01)$ and $h(10)$ has a length of at most $|h(01)| - 1$, since h is nonperiodic.

Lemma 5.18. *Let $z_h = h(01) \wedge h(10)$ and denote $m = |z_h|$. Then $h^{(m)}$ is a marked morphism and $h^{(m)}(w) = z_h^{-1}(h(w)z_h)$, for all $w \in \{0,1\}^*$. Moreover, for any w, if $|h(w)| \geq m$, then $z_h \leq h(w)$.*

Proof. We may assume, by symmetry, that $|h(1)| \geq |h(0)|$. Suppose first that $m < |h(0)|$. Then, clearly, $h^{(m)}(0)$ and $h^{(m)}(1)$ begin with different letters by the maximality of the z_h.

On the other hand, if $m \geq |h(0)|$, then we have two possibilities, either $m \leq |h(1)|$ or $m > |h(1)|$.

Assume first that $m \leq |h(1)|$. Then $h(10) = h(0)^k uv$ for some $k > 0$, $u, v \in \{0,1\}^*$ such that $z_h = h(0)^k u$. Also, $h(01) = h(0)^k uxw$, where $ux = h(0)$ and $w \in \{0,1\}^*$. Note that v and x begin with different letters by

the definition of z_h. Since $h(10) = h(0)^k uv$, we obtain that $h(1) = h(0)^k u \cdot v_1$, where $v = v_1 h(0)$. It follows that

$$h^{(m)}(1) = v_1 h(0)^k u = v_1 h(0) h(0)^{k-1} u = v h(0)^{k-1} u.$$

Also, since $h(0) = ux$ and $m \equiv |u| \pmod{|h(0)|}$, we obtain that $h^{(m)}(0) = xu$ (see also Figure 5.7).

Assume next that $m > |h(1)|$. Again, $h(10) = h(0)^k uv$ for some $k > 0$, $u, v \in \{0,1\}^*$ such that $z_h = h(0)^k u$ and $h(01) = h(0)^k ux$, where v and x begin with different letters. Clearly $ux \leq h(0)$, $h(010) = h(0) \cdot h(0)^k uv = h(0)^k ux \cdot h(0)$. Now $h(1) = h(0)^k uv h(0)^{-1}$, and therefore

$$h^{(m)}(1) = v h(0)^{-1} h(0)^k u = v h(0)^{k-1} u.$$

Since $h(0) = uxz$ for some $z \in \{0,1\}^*$ and $m \equiv |u| \pmod{|h(0)|}$, we obtain that $h^{(m)}(0) = xu$.

We have proved that $h^{(m)}(1) = v h(0)^{k-1} u$ and $h^{(m)}(0) = xzu$, where $z = \varepsilon$, if $m \leq |h(1)|$, and since $v \wedge x = \varepsilon$, $h^{(m)}$ is marked.

For the rest of the claim, since $h(0) = uxz$, we observe that

$$h^{(m)}(0) = xzu = (h(0)^k u)^{-1} h(0) h(0)^k u = z_h^{-1} h(0) z_h,$$

and, since $h(1) = h(0)^k uv h(0)^{-1}$,

$$h^{(m)}(1) = v h(0)^{k-1} u = (h(0)^k u)^{-1} h(0)^k uv h(0)^{-1} h(0)^k u = z_h^{-1} h(1) z_h.$$

Therefore for all $w \in A^+$, $h^{(m)}(w) = z_h^{-1} h(w) z_h$, and $h^{(m)}$ is a morphism. The last part of the claim follows directly from this. □

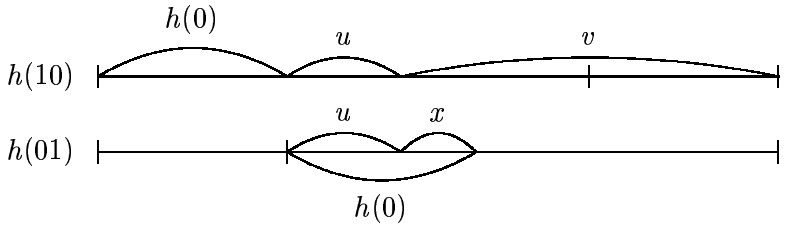

Figure 5.7: Case $k = 1$, $m < |h(1)|$.

Note that if h is already marked, then $z_h = \varepsilon$.

Let (h, g), where $h, g\colon \{0,1\}^* \to \{0,1\}^*$, be an instance of the binary PCP. Assume further that h and g are nonperiodic. Let z_h be as above, $m = |z_h|$ and $n = |z_g|$. We may assume, by symmetry, that $m \geq n$.

Lemma 5.19. *The instance (h, g) of the binary PCP has a solution if and only if the instance $((z_g^{-1} z_h, \varepsilon), h^{(m)}, g^{(n)}, (\varepsilon, z_g^{-1} z_h))$ of the binary GPCP has a solution.*

Proof. It is obvious that if an instance (h, g) of the PCP has a solution, then $z_g \leq z_h$, since $m \geq n$. Indeed, if w is a solution such that $n, m \leq |h(w)|$, then $z_h \leq h(w)$ and $z_g \leq g(w)$.

Assume first that the instance of the GPCP has a solution w, i.e.,

$$z_g^{-1} z_h h^{(m)}(w) = g^{(n)}(w) z_g^{-1} z_h.$$

Therefore,

$$z_g^{-1} h(w) z_h = z_g^{-1} g(w) z_h,$$

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

Assume then that (h, g) has a solution w. Since $h^{(m)}$ and $g^{(n)}$ are morphisms, we have

$$z_h(z_h^{-1} h(w) z_h) z_h^{-1} = h(w) = g(w) = z_g(z_g^{-1} g(w) z_g) z_g^{-1}$$

and therefore

$$z_h h^{(m)}(w) z_h^{-1} = z_g g^{(n)}(w) z_g^{-1}.$$

This is true if and only if

$$(z_g^{-1} z_h) h^{(m)}(w) = g^{(n)}(w)(z_g^{-1} z_h).$$

This proves the claim. □

We obtain the following corollary of Theorem 5.1,

Theorem 5.2. *The binary PCP is decidable.*

5.4.2 From binary GPCP to marked GPCP

Next we shall prove that the binary GPCP can also be reduced to the marked binary GPCP. This reduction is a bit more complicated than the reduction for the binary PCP. Also, for the GPCP we know that it is decidable if one of the morphisms is periodic, see Theorem 3.6. Again we assume that the morphisms are nonerasing.

Let $I = ((p_1, p_2), h, g, (s_1, s_2))$ be an instance of the binary GPCP, where $h, g\colon \{0,1\}^* \to \{0,1\}^*$ and $p_1, p_2, s_1, s_2 \in \{0,1\}^*$. Let z_h and z_g be as in Section 5.4.1 and $m = |z_h|$ and $n = |z_g|$. Assume again that $m \geq n$

Assume that I has a solution w, where $|h(w)| \geq m$ and $|g(w)| \geq m$. By Lemma 5.18, w has a suffix x such that $z_h \leq h(x)$ and $z_g \leq g(x)$. We are interested in the shortest of such suffixes.

Lemma 5.20. *Let $I = ((p_1, p_2), h, g, (s_1, s_2))$ be an instance of the binary GPCP. Then I has a solution w such that $|w| \geq \max\{m, n\}$ if and only if the instance $I' = ((p_1 z_h, p_2 z_g), h^{(m)}, g^{(n)}, (y_1 s_1, y_2 s_2))$ of the marked binary GPCP, where $z_h y_1 = h(x)$ and $z_g y_2 = g(x)$ for some word x, has a solution.*

Proof. Assume that $m \geq n$ and that I has a solution w, $|w| \geq m$. Now, by the above, w has a suffix x such that $h(x) = z_h y_1$ and $g(x) = z_g y_2$. Let $w = w'x$. Then

$$p_1 h(w) s_1 = p_1 h(w') h(x) s_1 = p_1 z_h z_h^{-1} h(w') z_h y_1 s_1 = p_1 z_h h^{(m)}(w') y_1 s_1,$$

and, similarly,

$$p_2 g(w) s_2 = p_2 z_g g^{(n)}(w') y_2 z_2.$$

Therefore w' is a solution of I'. This also proves the other direction, since the assumptions on x in I' yield that the above arguments can also be repeated by starting with w', and then achieving $w = w'x$. □

We are ready to prove the following theorem, which is actually a corollary of Theorem 5.1.

Theorem 5.3. *The binary GPCP is decidable.*

Proof. Let $I = ((p_1, p_2), h, g, (s_1, s_2))$ be an instance of the binary GPCP. If h or g is periodic, then I can be decided by Theorem 3.6. If this is not the case, then the decision can be made in the following way. The solutions w of I such that $|w| < m$ can be found by exhaustive search. For the solutions w such that $|w| \geq m$, we use Lemma 5.20 in the following way: generate the instance $I' = ((p_1 z_h, p_2 z_g), h^{(m)}, g^{(n)}, (y_1 s_1, y_2 s_2))$ of the marked binary GPCP, where $z_h y_1 = h(x)$ and $z_g y_2 = g(x)$, for all words x such that $|h(x)| \geq m$ and $|g(x)| \geq n$ and, for all $x' < x$, $|h(x')| < m$ or $|g(x')| < n$. If the images of h and g are nonempty, then there exist only finitely many such words x and we achieve only finitely many instances of the marked binary GPCP. Now the claim follows from Theorem 5.1. □

5.4.3 Equality sets of the binary PCP

Let us shortly consider the binary equality sets, i.e., the set of all solutions of an instance of a binary PCP. Let $I = (h, g)$, where $h, g\colon \{0,1\}^* \to \{0,1\}^*$, be an instance of the binary PCP. It is obvious that if both h and g are periodic, that the equality set $E(h, g)$ consists of words with a fixed ratio of letters 0 and 1. If only one of the morphisms is periodic, then $E(h, g) = u^*$, where u is the minimal solution.

The most interesting case is when both of the morphisms are nonperiodic. The following theorem is from Ehrenfeucht, Karhumäki and Rozenberg [6] (see also Choffrut and Karhumäki [4]).

Theorem 5.4. *Let h and g be nonperiodic binary morphisms. Then the equality set $E(h,g)$ is always of one of the following forms*

$$\{u,v\}^* \quad or \quad (uw^*v)^*$$

for some words $u,w,v \in \{0,1\}^*$.

Actually, it is not known whether the second case is possible. Therefore we state an important open problem:

Problem 5.1. *Is it true that the equality set of nonperiodic binary morphisms is always of the form $\{u,v\}^*$ for some words $u,v \in \{0,1\}^*$?*

Conclusions

Historically, the key problem of this work is the binary Post Correspondence Problem. It was stated as an exercise - without an asterix to indicate that the problem is difficult - in Hopcroft and Ullman [25] as well as in Harrison [24]. In both of these books it was expected that the problem has a positive answer, that is, that the binary PCP is decidable. In [25] this was stated explicitly and in [24] more implicitly. To confuse the issue more, Brainerd and Landweber [3] had it as an exercise and mentioned it to be open, but an undecidability result was expected. Even the PCP for three letter alphabets was included in the same exercise in [25].

The first solution for the binary PCP by Ehrenfeucht, Karhumäki and Rozenberg [5] was very complicated. Actually, to prove that the binary marked GPCP is decidable the authors used the idea of a successor to reduce the problem to specific cases where the proof could be finished by combinatorial case analysis. In this thesis we achieved a somewhat simpler proof essentially by using the concept of the successor iteratively. In fact, we were able to prove a much stronger result, namely, the decidability of the marked GPCP for arbitrary alphabets, not only for the binary ones.

Another central goal of this thesis was to search for the borderline of decidability and undecidability. We were able to prove that the borderline is between prefix and marked PCP, and, on the other hand, between 2-marked and 1-marked.

On the other hand, many problems related to the topic remain open. During the presentation of this work, in particular, the following open problems were encountered:

Problem 3.1. *Is the PCP decidable for domain alphabets of size n, when $3 \leq n \leq 6$?*

Problem 3.2. *Is the GPCP decidable for domain alphabets of size n, when $3 \leq n \leq 6$?*

Problem 4.1. *Is the strongly k-marked PCP decidable for $1 < k < 5$?*

Problem 5.1. *Is it true that the set of all solutions of the PCP for non-periodic binary morphisms is always of the form $\{u, v\}^*$ for some words $u, v \in \{0, 1\}^*$?*

Bibliography

[1] B. Baker and R. Book. Reversal-bounded multipushdown machines. *J. Comput. System Sci.*, 8:315–332, 1974.

[2] J. Berstel. *Transductions and Context-Free Languages.* B. G. Teubner, 1979.

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

[4] 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-Verlag, 1997.

[5] 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.

[6] A. Ehrenfeucht, J. Karhumäki, and G. Rozenberg. On binary equality languages and a solution to the test set conjecture in the binary case. *J. Algebra*, 85:76–85, 1983.

[7] S. Eilenberg. *Automata, Languages, and Machines*, Vol. A. Academic Press, 1974.

[8] S. A. Greibach. An infinite hierarchy of context-free languages. *J. Assoc. Comput. Mach.*, 16:91–106, 1969.

[9] S. A. Greibach. Remarks on blind and partially blind one-way multicounter machines. *Theoret. Comput. Sci.*, 7:311–324, 1978.

[10] V. Halava. Decidable and undecidable problems in matrix theory. Master's thesis, University of Turku, 1997.

[11] V. Halava. Finite substitutions and integer weighted finite automata. Licentiate thesis, University of Turku, 1998.

[12] V. Halava and T. Harju. Undecidability in integer weighted finite automata. *Fund. Inform.*, 34:189–200, 1999.

[13] V. Halava and T. Harju. Infinite solutions of the marked Post Correspondence Problem. In W. Brauer, H. Ehrig, J. Karhumäki and A. Salomaa, editors, *Formal and Natural Computing. Lecture Notes in Comput. Sci.* 2300, pages 57–68. Springer-Verlag, 2002.

[14] V. Halava and T. Harju. Mortality in matrix semigroups. *Amer. Math. Monthly*, 108(7):649–653, 2001.

[15] V. Halava and T. Harju. Some new results on Post Correspondence Problem and its modifications. *Bull. of the EATCS*, 73:131–141, 2001.

[16] V. Halava and T. Harju. An undecidability result concerning periodic morphisms. Technical Report 397, TUCS, 2001. To appear in Lecture Notes in Comput. Sci. 2295.

[17] V. Halava, T. Harju, and M. Hirvensalo. Generalized PCP is decidable for marked morphisms. *Lecture Notes in Comput. Sci.* 1684, pages 304–315. Springer-Verlag, 1999.

[18] V. Halava, T. Harju, and M. Hirvensalo. Binary (generalized) Post Correspondence Problem. Technical Report 357, TUCS, 2000. To appear in Theoret. Comput. Sci.

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

[20] V. Halava, M. Hirvensalo, and R. de Wolf. Decidability and undecidability of marked PCP. *Lecture Notes in Comput. Sci.* 1563, pages 207–216. Springer-Verlag, 1999.

[21] V. Halava, M. Hirvensalo, and R. de Wolf. Marked PCP is decidable. *Theoret. Comput. Sci.*, 255(1-2):193–204, 2001.

[22] T. Harju and J. Karhumäki. Morphisms. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 1. pp. 439–510, Springer-Verlag, 1997.

[23] T. Harju, J. Karhumäki, and D. Krob. Remarks on generalized Post Correspondence Problem. *Lecture Notes in Comput. Sci.* 1046, pages 39–48. Springer-Verlag, 1996.

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

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

[26] O. H. Ibarra. Restricted one-counter machines with undecidable universe problems. *Math. Systems Theory*, 13:181–186, 1979.

[27] Y. Lecerf. Récursive insolubilité de l'équation générale de diagonalisation de deux monomorphisms de monoïdes libres $\varphi x = \psi x$. *C. R. Acad. Sci. Paris*, 257:2940–2943, 1963.

[28] M. Lipponen. *On Primitive Solutions of the Post Correspondence Problem.* PhD thesis, University of Turku, 1996.

[29] L. P. Lisovik. An undecidable problem for countable Markov chains. Cybernetics, 27 2:163–169, 1991.

[30] Y. Matiyasevich. Enumerable sets are diophantine. *Soviet Math. Doklady* 11:354–357, 1970 (English transl. *Dokl. Akad. Nauk. SSSR* 191:279–282, 1971)

[31] Y. Matiyasevich and G. Sénizergues. Decision problems for semi-Thue systems with a few rules. In *Proceedings, 11$^{\text{th}}$ Annual IEEE Symposium on Logic in Computer Science*, pages 523–531, New Brunswick, New Jersey, 27–30 July 1996. IEEE Computer Society Press.

[32] M. S. Paterson. Unsolvability in 3×3 matrices. *Stud. Appl. Math.*, 49:105–107, 1970.

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

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

[35] G. Rozenberg and A. Salomaa. *Cornerstones of Undecidability.* Prentice Hall, 1994.

[36] K. Ruohonen. Reversible machines and Post's correspondence problem for biprefix morphisms. *Elektron. Informationsverarb. Kybernet. (EIK)*, 21(12):579–595, 1985.

[37] A. Salomaa. *Formal Languages.* Academic Press, 1973.

[38] G. C. Tzeitin. Associative calculus with an unsolvable equivalence problem. *Tr. Mat. Inst. Akad. Nauk*, (52):172–189, 1958. (In Russian)